14 (1)

THE DESIGN OF A TAPE MACROMODULE

AD668964

TECHNICAL REPORT No. 3

JUNE 15 1967

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151

Computer Research Laboratory
Washington University
St. Louis, Missouri

DDC
RECEIVED
MAY 16 1968
C

This document has been approved for public release and sale; its distribution is unlimited.

96

BEST
AVAILABLE COPY

THE DESIGN OF A TAPE MACROMODULE

Warren M. Littlefield

June 15, 1967

Computer Research Laboratory
Washington University
724 South Euclid Avenue
St. Louis, Missouri

This work has been supported by the Advanced Research Projects Agency of the Department of Defense under contract SD-302 and by the Division of Research Facilities and Resources of the National Institutes of Health under Grant FR-00218.

"Why Not!"

Wesley A. Clark (1965)

PREFACE

This thesis deals with the design and function of a Magnetic Tape System Module. The prototype was built out of MECL integrated logic used in the pulse manner, and handles all those functions peculiar to a basic LINC[1] tape transport. The system was interfaced and debugged on a LINC which was programmed to behave as a Macromocular interface. In order to achieve a complete understanding of the material contained in this thesis, it is necessary that one be thoroughly familiar with the general theory of tape operation as contained in the first six chapters of "LINC Theory of Operations II, Volume 14"[2].

I would like to thank Wesley A. Clark and William N. Papian of the Computer Research Labs of Washington University for making this project possible. Also, Severo M. Ornstein and Charles E. Molnar whose original work on the LINC tape system was of immeasurable help to me. My deepest appreciation goes to Thomas J. Chaney whose design of the critical timing elements was of fundamental importance. Finally, I am indebted to George Perry, Norman Kinch, and my wife, Marjorie, for their help in the preparation of this manuscript.

---

[1]Wesley Clark and Charles Molnar, "The LINC", Proceedings, New York Academy of Sciences Conference on Computers in Medicine and Biology, May 27-29, 1963.

[2]Severo Ornstein and Charles Molnar, "LINC Theory of Operations II", (St. Louis: Washington University, 1967), XIV.

TABLE OF CONTENTS

Title Page	i
Preface	iii
Table of Contents	iv
List of Tables	v
List of Figures	vi
List of Drawings	vii
Introduction	1
Marking Code	6
Data Channel Structure	9
Search Phase	11
Read-Check Phase	14
Write Phase	16
Motion Option	18
Mark Phase	19
State Flipflops	22
Data Registers	25
Timing Logic	27
Electrical Considerations	29
Testing Procedure	32
Economics	43
Appendix I: Instruction Code	45
Appendix II: Logic Conventions	48
Appendix III: Design of Timing Circuits	58
Appendix IV: Magnetic Tape Program	64
Appendix V: Marking Tape Program	66
Appendix VI: Schematics	71
Bibliography	78

LIST OF TABLES

I.	Marking Table	6
II.	Motion State Table	23
III.	Coding Table	23
IV.	Cost Tabulation	43

LIST OF FIGURES

1.	Tape Module	2
2.	ADC Cards	3
3.	Marking Code	7
4.	Data Flow	
5.	Tape Channel Layout	10
6.	SCH Phase Timing Diagram	12
7.	RCH Phase Timing Diagram	12
8.	WRT Phase Timing Diagram	12
9.	Mark Phase Timing Diagram	12
10.	Test Assembly Layout	32
11.	Magnetic Tape Instruction	34
12.	Marking System	35
13.	Software Marking Routine	36

LIST OF DRAWINGS

1.	State Flipflops	71
2.	Decoders	72
3.	Motion Control	73
4.	Mark, Timing and Delays	74
5.	DEC to MECL Buffers	75
6.	MECL to DEC Buffers	76
7.	Tape Register	77

BLANK PAGE

INTRODUCTION

The Magnetic Tape System was built in order to see what the functional requirements of an input-output Macromodule were, and precisely what logic network was essential to create such an autonomous unit. It is hoped that this machine will form the prototype for the eventual Tape Macromodule which will be compatible with the Macromodular computing system being developed at the Computer Research Labs. In any case, this system will have crystallized the definition of what is necessary to effectively utilize a LINC tape transport, and interface it to the computational world.

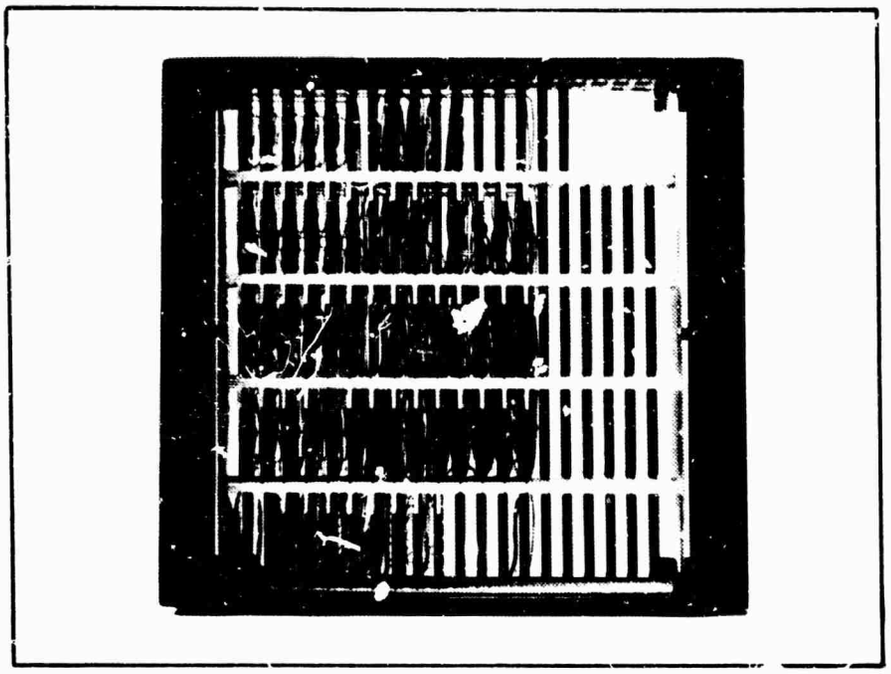
The basic functional philosophy of the Tape Module was derived from the LINC tape system[3], as its functional utility has been time-proven. The block storage format of data words has remained the same. The Marking code has been changed to simplify the logic and eliminate the spurious codes which must be accounted for. Pulse logic rather than transition logic was chosen as it is more amenable to the magnetic tape format. The operational codes consisted of single block commands only.

Much logic redundancy was used in order to avoid any catastrophic hazards. However, as this is an input-output device, and thus subject to "real world" noise and "glitch" problems, error testing logic and failure modes have been supplied. The emphasis was placed on safety rather than on logical economy.

---

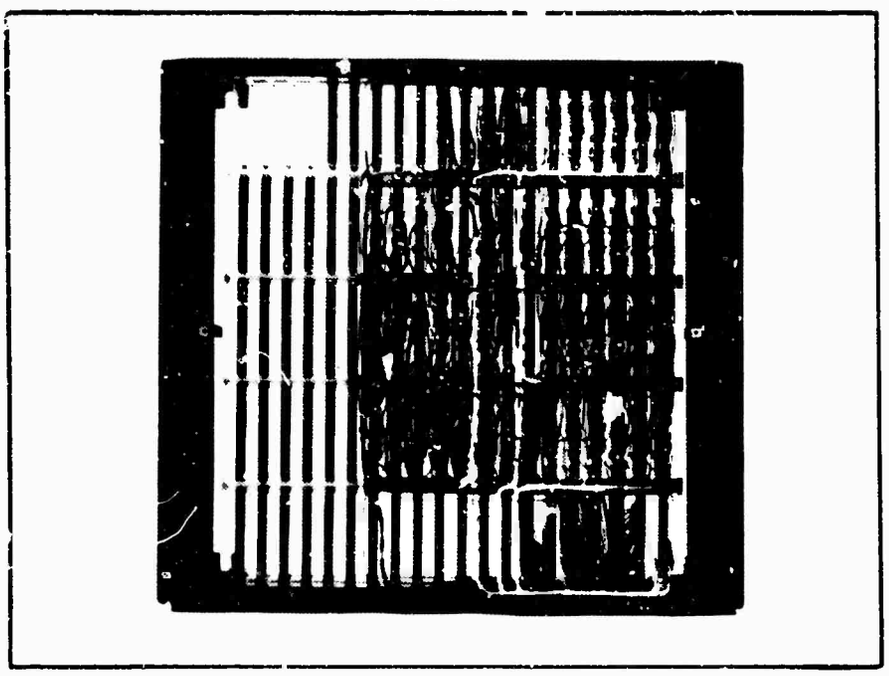
[3] Ibid.

Figure 1:



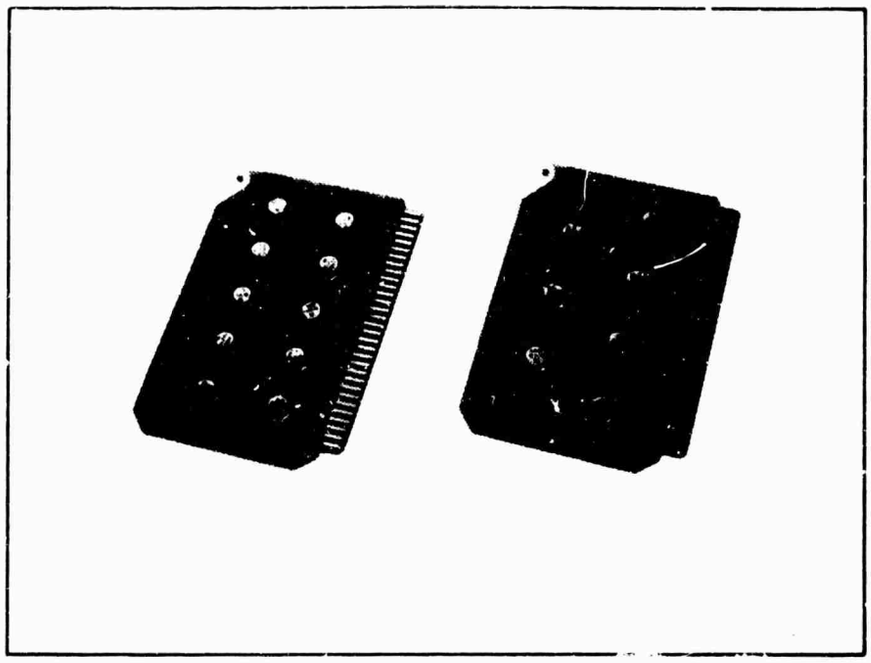
Front

Back

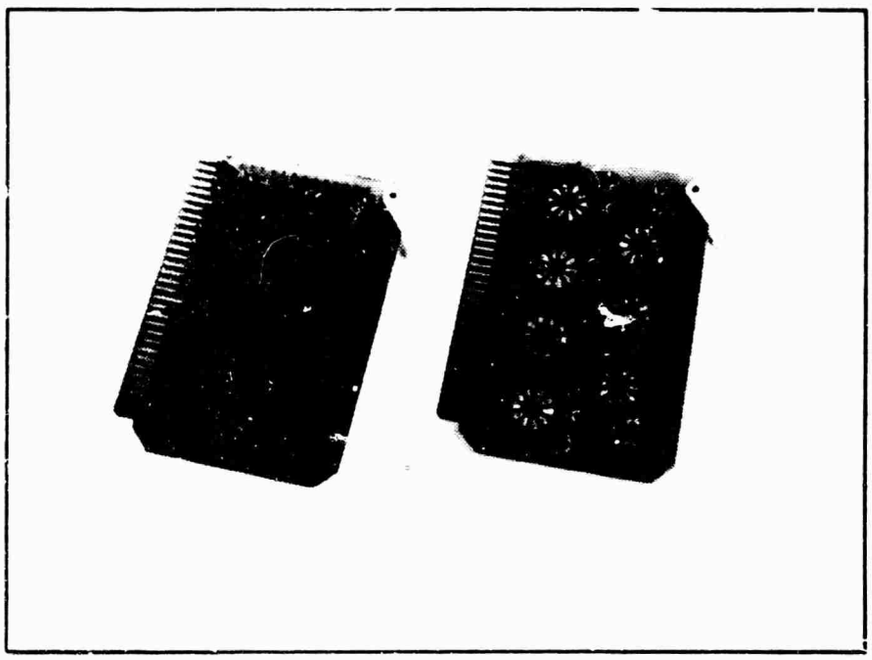


Tape Module

Figure 2:



General Pattern Cards General Purpose Cards



ADC Cards

The system consists of 335 logic "cans" of the Motorola MC350 series. These devices are emitter-coupled integrated circuits and were chosen because of their high fan-in, fan-out, reliability, and availability. The cards used were Applied Development Corporation General Purpose, and General Pattern cards. The General Purpose cards have 62 channels, with common power, and common ground lines; whereas the General Pattern have only 31 channels available, but may be wired in any manner. The frame was made large enough to accomodate any future changes or additions.

System operation is initiated with a "DO" pulse which starts the motion of the machine, and also loads the Block number, instruction code, unit number, and motion option into the module from the appropriate windows of the outside system. After the system has accelerated the tape up to operational speed, it searches for the correct block, and then proceeds to execute the instruction. After completion of any tape instruction, the module will indicate the execution with a DONE level. If it has failed in the transferal of information, a Fail flag will also be set. After completion of the instruction, the system will regard the Motion Option (i-bit), and will either reverse and halt tape motion, or allow it to proceed in the same direction that it was travelling when the execution was completed.

All timing pulses used by the system during Magnetic Tape Instructions are generated from the timing track of the tape. The relative position of the tape is indicated by the Mark track of the tape. As all operations are done in blocks of data, each operation is preceeded by a Search phase in which the appropriate Block is located.

Provision has been made to initially Mark blank tapes so that they conform to the pattern utilized by this system. During the Mark phase, the system is placed in the Marking mode which causes the module to run on its own internal Mark clock, rather than on tape time. During this phase only are the Timing and Mark tracks as well as the Data tracks affected. During all other Magnetic Tape phases, only the Data tracks can be altered.

MARKING CODE

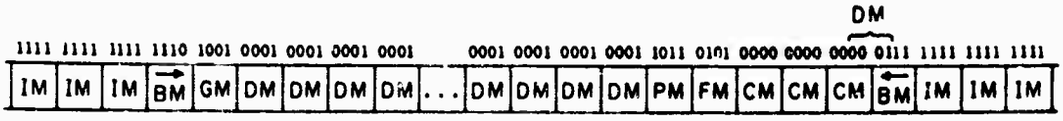
The marking code used on this tape system is unique, and incompatible with those codes presently used on other tape-controlled tape systems. This code has the advantages of simplicity and nonambiguity. It contains all those features which have been time-proven on the LINC, and does away with those features which have not been found worthwhile, but because of their presence have resulted in ambiguity and the additional hardware needed to resolve this ambiguity.

The tape is broken down into blocks of information. These blocks can be of any length and there can be up to $10{,}000_8$ blocks on the tape (if they are short enough to physically fit on the tape). The present tapes used by this system employ block lengths of 400_8 words, with 1000_8 blocks on the tape. There are eight distance codes used in each block.

Table 1:

Mark Code	Abbreviation	Description
1111	IM	Interblock Zone Mark, Data channels unused.
1110	$\overrightarrow{BM}$	Forward Block Mark, Data channels contain Fwd. Blk. #.
1001	GM	Guard Mark, Data channels unused.
0001	DM	Data Mark, Data channels contain data words.
1011	PM	Pre-Final Mark, Data channels contain data word.
0101	FM	Final Mark, Data channels contain data word.
0000	CM	Check Mark, Data channels contain check-sum.
0111	$\overleftarrow{BM}$	Backwards Block Mark, Data channels contain Bwd. Blk #.

Figure 7:



Beneath the forward block mark is placed the forward block number. In reality, it is the complemented forward block number that is stored in the data channels. Likewise, beneath the backwards block mark is stored the backwards block number. This number is stored as a complemented reversed block number (see Figure 5).

The function of the guard mark (GM) is to supply a buffer zone between the block number and the data words. This is done so that when the Writers are turned on at the end of a GM, the block number won't be gaussed.

Beneath the data, pre-final, and final marks, all the data words are written. The need for a PM and an FM is obvious if one looks at a WRT instruction. As soon as all the data has been written on tape, it is necessary to write the check-sum which is stored in a different register than the data words. Thus, it is necessary to know when to change registers. Likewise, it is necessary to anticipate the FM with a PM since the system is using a double ranking data transfer. This is so that the system does not ask for more data words than it intends to use to complete the block. Hence, during a PM or FM, the WRTPRDY flipflop is not turned on again.

The function of the Check mark is to 1) turn off the writers; and 2) indicate the presence of a check-sum. Thus, beneath the CM is stored the one's complement of the correct check-sum for that block. There are three CM's at the end of each word to 1) insure that the Writers will be turned off; and 2) act as a buffer zone so that the backwards block number does not get gaussed. The CM code was chosen so that it is adjacently redundant. Thus, as you pass through the Check zone you do not see three CM's, but ten (i.e., it is possible to get ten sets of four adjacent 0's out of a row of thirteen 0's).

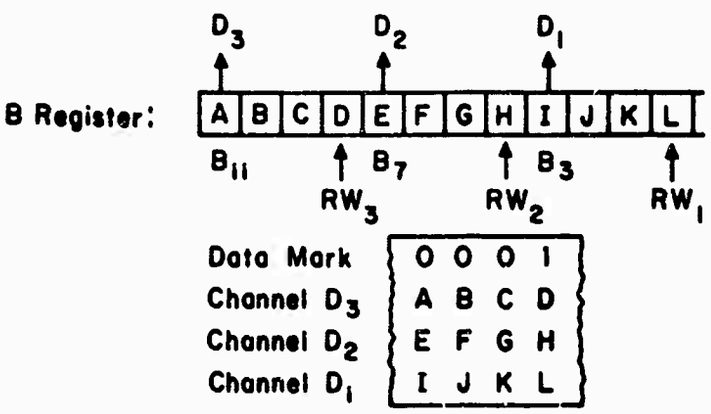
The function of the backwards block mark is to supply a block number when the tape is traveling in the backwards direction. The $\overleftarrow{\text{BM}}$ and the IM are the only marks that are utilized when the tape is in reverse motion.

Note that there is only one spurious IM in the entire block, and it occurs in a place where it cannot possibly do any harm, as it will only be seen in the SEARCH phase.

DATA CHANNEL STRUCTURE

If we lable each of the bits alphabetically, starting with the most significant bit, Figure 4 shows how the information is transformed from the B register to the data tracks.

Figure 4:



The following demonstrates how block number 235_8 would be written on tape below a forward block number, and below a backwards block number. Decoding 235_8 in binary: 000 010 011 101. If we arrange this in hexadecimal form it becomes obvious how this is arranged on tape: 0000 1001 1101. Remember that it is the one's complement of the block number that is stored on tape, hence, the number we will find is: 1111 0110 0010. The backwards block number is more difficult to visualize. Two transformations must be made. First, there must be a polarity inversion because of direction change, and second, there must be a reversal in bit order as we are now reading from right to left instead of from left to right. If we take the

complement again, the polarity will be correct: 0000 1001 1011. Then we must reverse the order of each four bits: 0000 1001 1011. Note that because of symmetry, only the last four bits appear to have changed. Figure 5 shows how this appears on tape.

Figure 5:

BLOCK 235_8

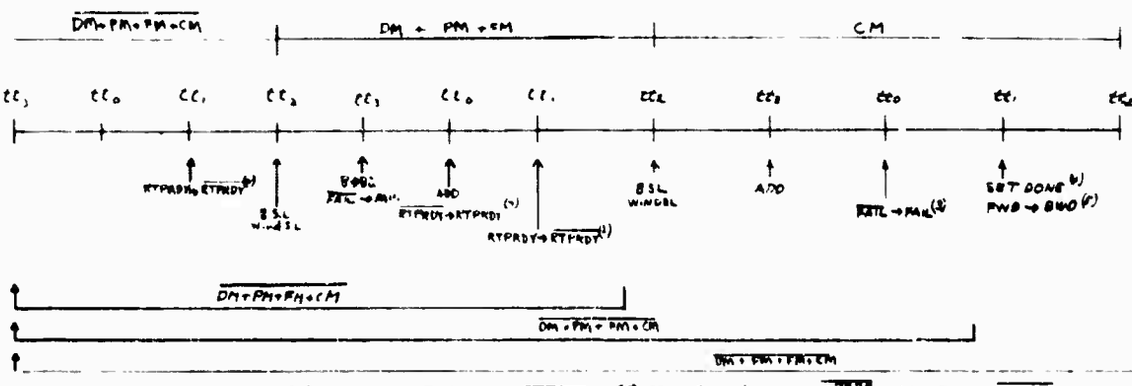
Fwd. Block Mark	1	1	1	0	Bwd. Block Mark	0	1	1	1
Channel D_3	1	1	1	1	Channel D_3	0	0	0	0
Channel D_2	0	1	1	0	Channel D_2	1	0	0	1
Channel D_1	0	0	1	0	Channel D_1	1	0	1	1

SEARCH PHASE

Magnetic tape operations are initiated by a Magnetic Tape Pulse (MTP) which immediately places the system in the Search Phase (see timing chart on Figure 6). The MTP will set all state flipflops in the system, hence, negating any operation the system may be executing previous to the MTP. MTP sets the Search flipflop to logical 1, loads the S register, the C register, the motion option (i-bit), the unit number, and sets the motion flipflops, while clearing all remaining state flipflops. If initially at rest, the Motion flipflops are put in the 1 state. If not at rest, the motion remains unaffected.

Consulting the SEARCH timing diagram (see Figure 6), note that unless a Block Mark is in the window register, the A register will be alternately cleared and loaded. As all window shifting is done at tt_2 time, the first pulse that will recognize a new mark is tt_3. Since the A register is cleared at tt_3 time, and loaded at tt_0 time, when the Block Mark finally does appear, the A register will be loaded with the desired block number. At $BM \cdot tt_3$, the complemented block number of the block just passing on tape will be added to the desired block number in 2's complement manner. If the resulting sum is all 1's (or in octal form, all 7's) then the correct block has been located, in which case at $BM \cdot tt_0$ the Search flipflop is cleared, and the system proceeds from the Search phase to the instruction phase. If the instruction to be executed is a write instruction (WRT), then at $BM \cdot tt_0 \cdot A^{7777}$, the write tape ready flipflop (WRTPRDY) is turned on. The Search phase is completed with the clearing of the A register. If the addition did not result in all ones ($A^{\overline{7777}}$), then the Search flipflop is

RCH



(1) RCH (DM+PM+FM)·TPRDY·tt$_3$ ⊃ $\overline{\text{Fail}}$ → Fail

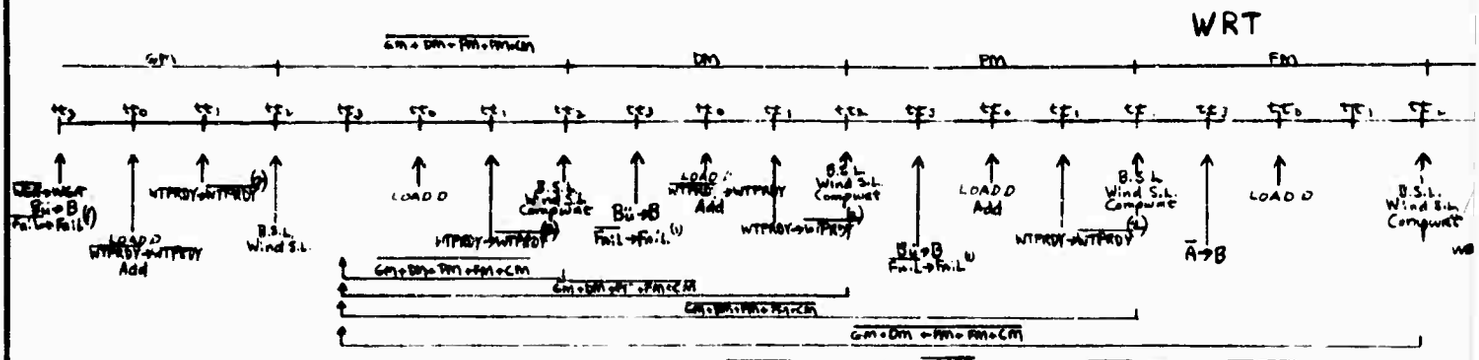
(2) OK·tt$_1$ ⊃ RTPRDY → $\overline{\text{RTPRDY}}$

(3) ROC·Search·CM·$\overline{A^{7777}}$·tt$_0$ ⊃ $\overline{\text{Search}}$ → Search
RCH·Search·CM·$\overline{A^{7777}}$·tt$_0$ ⊃ $\overline{\text{Fail}}$ → Fail

(4) $\overline{\text{Search}}$

(7) (DM+PM+FM)·RTP·tt$_0$ ⊃ $\overline{\text{RTPRDY}}$ → RTPRDY

WRT



(1) WAT·(GM+DM+PM)·WTPRDY·$\overline{\text{Search}}$·tt$_3$ ⊃ $\overline{\text{Fail}}$ → Fail

(2) OK·tt$_1$ ⊃ WTPRDY → $\overline{\text{WTPRDY}}$, Load Bü

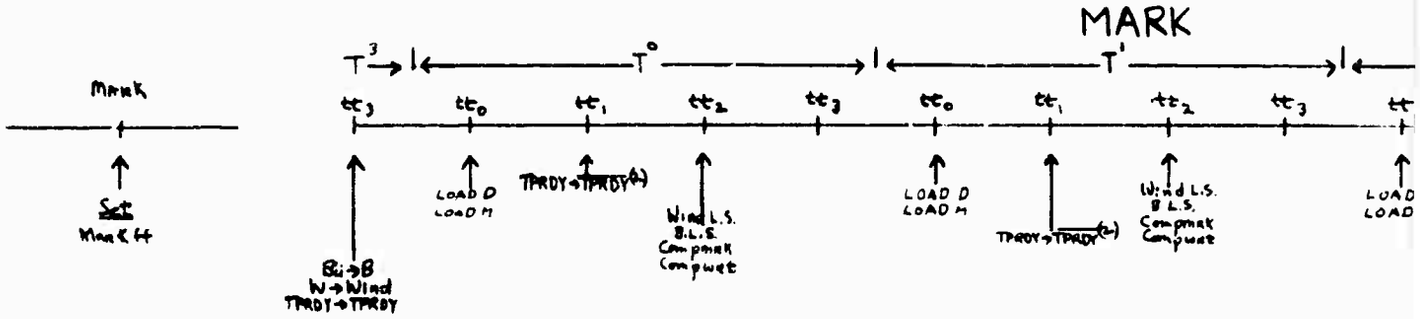
(3) tt$_0$·CM·$\overline{\text{Search}}$·WRC ⊃ Comp Check, Check ⊃ Search

(4) $\overline{\text{Search}}$·CM·tt$_0$·$\overline{A^{7777}}$·Check ⊃ $\overline{\text{Fail}}$ → Fail, $\overline{\text{Search}}$ → Search

(5) $\overline{\text{Search}}$·$\overline{\text{Check}}$·$\overline{A^{7777}}$·tt$_1$·CM ⊃ Set done

(6) CM·i·$\overline{\text{Search}}$·tt$_1$ ⊃ FWD → BWD

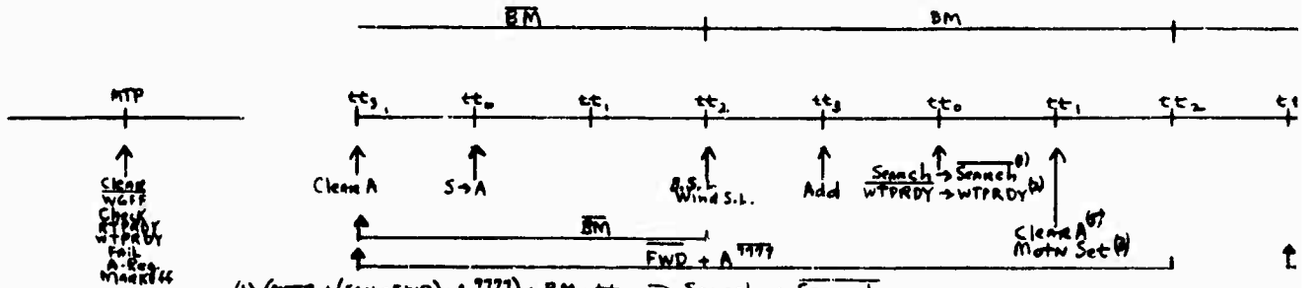
MARK



(1) Mark ff·T^3·TPRDY·tt$_2$ ⊃ $\overline{\text{Fail}}$ → Fail

(2) OK·tt$_1$ ⊃ TPRDY → $\overline{\text{TPRDY}}$, Load B, Load W

SEARCH



(1) (MTB+(SCH+FWD)·$\overline{A^{7777}}$)·BM·tt$_0$ ⊃ Search → $\overline{\text{Search}}$

(2) WRT·BM·$\overline{A^{7777}}$·FWD·tt$_0$ ⊃ $\overline{\text{WTPRDY}}$ → WTPRDY

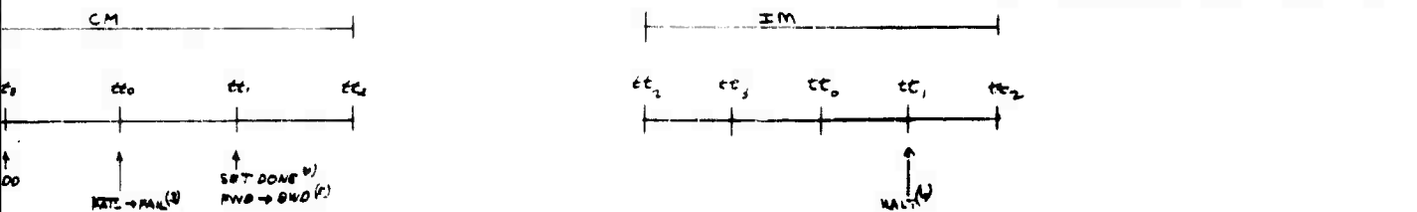
(3) Search·A$'_H$ ⊃ Set to BWD; Search·A$^0_{11}$ ⊃ Set to FWD

(4) OK·tt ⊃ TPRDY → $\overline{\text{TPRDY}}$, Load Bü

(5) BM·$\overline{\text{Search}}$·tt$_1$ ⊃ clear A'

RCH

RCH = RDE + RDC · WRC · CHECK + CHK

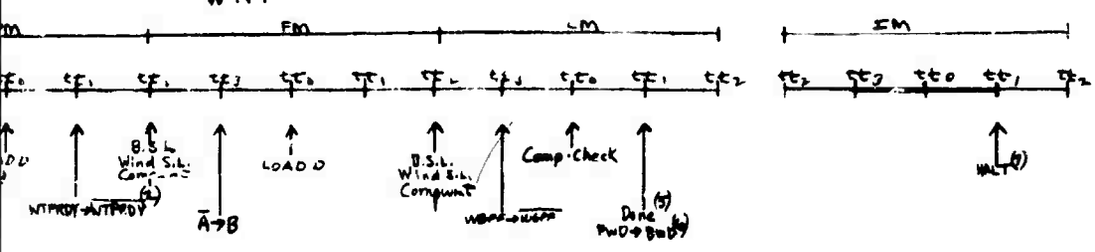


(4) $\overline{Search} \cdot CM \cdot \overline{Mark\ FF} \cdot tt_1 >$ Set done (5) $CM \cdot i \cdot \overline{Search} \cdot tt_1 >$ FWD → BWD (6) $IM \cdot i \cdot \overline{Search} \cdot tt_1 >$ HALT

$CM \cdot \overline{A^{7777}} \cdot tt_0 >$ $\overline{Search} \rightarrow$ Search
$CM \cdot \overline{A^{7777}} \cdot tt_0 >$ Fail → Fail

WRT

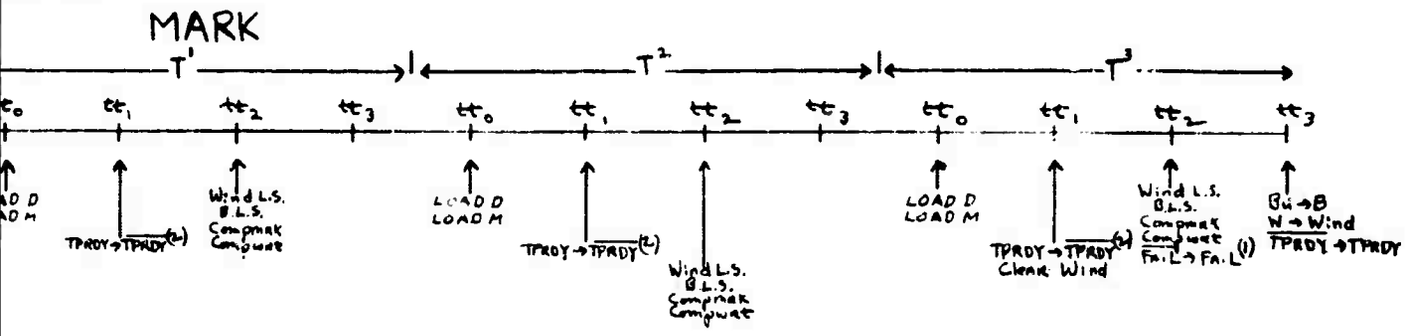
WRT = WRI + WRC · $\overline{CHECK}$



(1) $IM \cdot i \cdot \overline{Search} \cdot tt_1 >$ Halt

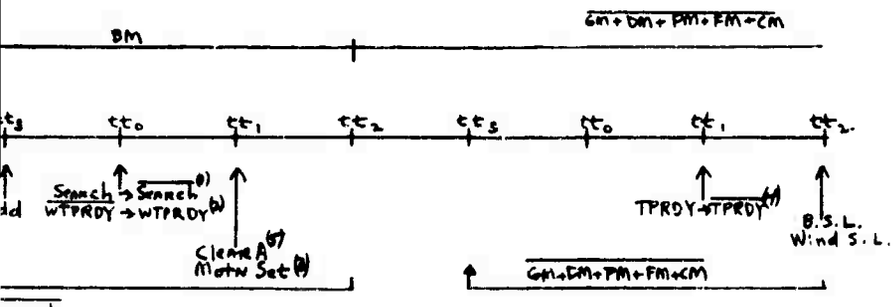
Check > $\overline{Fail}$ → Fail, $\overline{Search}$ → Search
$tt_1 \cdot CM >$ Set done
FWD → BWD

MARK



SEARCH

SEARCH = MTB + SCH + Search



arch
PRDY
Set to FWD

B

COMPUTER RESEARCH LABORATORY WASHINGTON UNIVERSITY ST. LOUIS, MISSOURI			
TITLE: TIMING DIAGRAM			
DATE	CHK: W. H. L	DRAWING NUMBER: 0	REV

BLANK PAGE

not reset, and the motion flipflops are examined to see if the tape is proceeding in the correct direction. If the result in the Accumulator was negative, then the tape has passed the desired block and the motion is placed in the backward direction. This motion change automatically fires the Acceleration in Progress Delay (ACIP) and the window register is cleared until the tape is up to speed in the backward direction. The window register will continue to be cleared until the Tape Time OK (TTOK) level stays on continuously, hence leaving a Check Mark in the window. If the accumulator remains positive after the addition, then the tape is proceeding in the correct direction and no change is made. In either case, the system continues in the SEARCH phase and repeats the preceeding.

In the case of the MTB instruction only, the Search phase is terminated after one block has been examined, and the system then proceeds to the motion option. Since the code for the MTB instruction is identical to the Done level, the machine will indicate that it has terminated the instruction as soon as the MTP has been received by the system.

The SCH instruction terminates execution upon the location of the desired block, and then immediately proceeds to the motion option, indicating completion upon passing the CM in the desired block. The MRK-enable instruction level is executed as a SCH instruction except that the tape motion will always be left in the forward direction if the i-bit option is 1. All other instructions proceed into the instruction phase upon completion of the SEARCH phase.

READ - CHECK PHASE

The RCH (Read or Check) phase encompasses all those operations wherein the tape is being interrogated and information is being taken from, rather than being placed on the tape. The RCH phase immediately proceeds after the completion of the SEARCH phase. It is properly defined by: RCH = RDE + RDC + WRC • CHECK + CHK. If the operation is a RTP (a RDC or RDE) instruction, then at $DM \cdot tt_0$ or $PM \cdot tt_0$ or $FM \cdot tt_0$, the RDTPRDY (Read Tape Ready) flipflop is turned on which tells the central computer that a data word is available in the $\ddot{BU}$ register. The central machine may then strobe in the word off the output data lines from the tape system. Upon receipt of the data word, the central machine turns on an OK level. At each tt_1, this level is examined, and if OK is asserted, the RTPRDY flipflop is cleared. If at $DM \cdot tt_3$ or $PM \cdot tt_3$ or $FM \cdot tt_3$ the RTPRDY level is still up, then a synchronous failure has occured, and the Fail flipflop is set. In any case, at that time the complete data word, now formed in B, is shifted to the $\ddot{BU}$ register for transferal.

For any RCH phase instruction, the current word in the B register is added to the accumulator at $DM \cdot tt_0$ or $PM \cdot tt_0$ or $FM \cdot tt_0$, thus forming the check-sum. At $CM \cdot tt_3$ the complemented check-sum from the tape is added to the accumulator. If the resulting sum is all ones (or all octal 7's, i.e., A^{7777}), the data transfer is assumed to have been correct. However, if at $CM \cdot tt_0$, the accumulator is not all ones ($A^{\overline{7777}}$), the Fail flipflop is turned on. If the instruction was a Read and Check (RDC) instruction, or the Check mode of a Write and Check (WRC) instruction, the system is put back into the Search phase, and the entire instruction is executed again. This

continues until successful completion is accomplished. However, if this is simply a Read (RDE) or a Check (CHK) instruction, the done level is set at CM·tt_1 time, regardless of the state of the Fail flipflop. Upon completion of the instruction, the motion option is executed.

WRITE PHASE

The Write phase (WRT) of operation is defined by either a WRI instruction or a WRC instruction in the $\overline{\text{CHECK}}$ mode. Locating the correct block terminates the Search phase and also turns on the Write Tape Ready (WRTPRDY) flipflop. This indicates to the main machine that the tape system is ready for the first word to be sent over. At $BM \cdot tt_1$, the accumulator is cleared so that the check-sum may be formed.

The Write Gate Flipflop (WGFF) is not turned on until $GM \cdot tt_3$. This turns on the writers and begins to flux the magnetic tape. At every tt_1 during a WRT instruction, the system looks to see if the central machine has turned on an OK level which says that the information is on the data lines waiting to be strobed in. If the OK level is up, then WRTPRDY is turned off and the information is strobed into the $\ddot{BU}$ register. If at tt_3 of any recognized mark, the WRTPRDY level is still up, then a synchronization failure has been made, and the Fail flipflop is turned on. At this time, the contents of the $\ddot{BU}$ register are shifted into the B register for transferal onto tape. At tt_0 of a GM, DM, or PM, the current data word in the B register is added to the contents of the A register in order to form the check-sum. At each tt_0 during a write, the writer flipflops of the D register are loaded from the 3, 7, and 11 bits of the B register. At tt_2 the D register is complemented causing a flux reversal to be recorded on tape, thus causing the information in the D register to be recorded on tape. At the same time, the B register is shifted left one bit, thus enabling the information to be serially read into the D register and onto tape. At $BM \cdot tt_0$ or $GM \cdot tt_0$ or $DM \cdot tt_0$, the WRTPRDY level is turned on again,

indicating that the BÜ register is empty and available for a new data word. The PM (Pre-Final Mark) is needed so as to anticipate the Final Mark (FM) during a write instruction. Since by the time the Pre-Final Mark has arrived, all the information that is to be written in that block of tape has been received by the tape system, it is not desirable to turn on the WRTPRDY level once again (thus requesting spurious information). At $FM \cdot tt_3$, the complement of the check-sum formed in the A register is shifted into the B register for recording on tape. This word is placed underneath the CM (Check Mark), which directly follows the last data word under the FM. At $CM \cdot tt_3$ the WGFF (Write Gate Flipflop) is cleared. Three Check Marks have been placed in series at the end of a word, and the code has been chosen so that $CM \cdot tt_3$ will appear ten times at the end of each block, virtually guaranteeing that the WGFF will be turned off before it reaches the backwards Block Mark.

If the instruction being executed is a WRC (Write and Check) instruction, the Check flipflop is turned on at $CM \cdot tt_0$, thus putting the machine in the Check mode. The check mode is discussed in the RCH Phase. If there has been no failure in a WRC and the Search flipflop is not on, than at $CM \cdot tt_1$ the Done level will be turned on by setting all the bits of the C register to 1, and the system will execute the motion option stored in the i Flipflop.

MOTION OPTION

If the i Flipflop is 0, then after completion of an instruction (DONE level is on) the motion flipflops will be complemented at CM·tt$_1$ time. This will fire the ACIP delay and reverse the direction of the tape. When the ACIP delay has run out, and ITOK is on continuously, the motion flipflops will be cleared at IM·tt$_1$. If the ACIP delay is set properly (about 110ms), this should stop the tape in front of the block that it just finished operating on. Thus, if a second operation is to be made on the same block by the time the tape is up to speed, the first block it should examine will be the desired block. The 0 motion option places the tape in the optimum position for recurrent block operations. The 1 motion option simply leaves the tape running in the same direction as it was when the operation was completed.

MARK PHASE

The Marking Phase is an entirely different mode of operation, as the system is run by an internal clock rather than by the timing track on the tape. The purpose of this phase is to place the Timing and Mark channels on the blank tape so that the tape may then be utilized as described in the other phases. The Mark Phase is initiated by a Mark pulse sent from the central machine. However, this Mark pulse will have no effect unless the Mark-enable command (MRK) is in the instruction window. The MRK-enable instruction is a safety lock placed on the machine so that accidental Marking will not occur, as the process of Marking entirely destroys any information that may previously have been stored on tape. With the MRK-enable in the C register, the MarkFF (Mark Flip-flop) is placed in the one state by the Mark pulse. The MarkFF enables the Mark clock and the Timing and Mark Reader Writer cards to function in the normal mode. The Mark pulse also puts the Motion flipflops in the Forward (FWD) state, thus starting the tape moving. Since a complete tape Mark is four bits long, with two tape-time pulses (t_i) per bit, a 2^3 counter is necessary to complete one full MARK cycle. T^0, T^1, T^2, and T^3, refer to the four bit cycles of the Mark, and reflect the four states of the two most significant bits of the T register, T_3 and T_2. The Mark Clock, T_0, simply fires a one-shot creating the Timing pulses, while T_1 gates them as either a tt_0 or a tt_2. The tt_1 pulse is created by the tt_0 pulse delayed 2 microseconds. Likewise, the tt_3 pulse is created by the tt_2 pulse delayed 2 microseconds. The basic Mark clock is a 20 microsecond clock, which is fed directly into T_1. Each stage of the T register counts down by a factor of two, thus creating a 160 microsecond period for a tape mark or a data word.

The process begins at $T^3 \cdot t_2$ with the turning on of the WRTPRDY flipflop. The central machine has 160 microseconds to respond to this level by placing on the data lines the tape Marks and data words that it wishes written next. When the data lines are set, the central machine turns on an OK level which the tape system examines every tt_1. If $OK \cdot tt_1$ asserts, then the WRTPRDY flipflop is turned off, and the information is strobed into the W register (tape Mark) and the $\ddot{BU}$ register (data word), thus completing the data transfer cycle. If however, the TPRDY level is still up at $T^3 \cdot tt_2$, thus indicating that the OK level never appeared, then the Fail flipflop is set to one. At $T^3 \cdot tt_3$, the new information stored in the W and $\ddot{BU}$ registers is shifted into the Wind and B registers respectively. At tt_0, the contents of $Wind_3$, B_{11}, B_7, and B_3, are loaded into the Mark and Data flipflops of the D register for writing onto tape. At tt_2, the D register is complemented placing a flux reversal of the correct polarity on tape. At the same time, the Wind and B registers are shifted left one bit. This process continues through all four bits of a MARK cycle for as long as the MRK level is asserted. It is the responsibility of the central machine to insure that the correct information is supplied to the tape system for placing on tape as any arbitrary data and tape Marks may be written in this manner. The marking format that is recognized by this particular system is discussed in the Marking Code section.

The tape Timing tracks are generated by the T_1 flipflop of the T register. Thus, when $\overline{T}_1$ is true, the Mark cycle is in the tt_0 and tt_1 segment, while when T_1 is true, the Mark cycle is in the tt_2 and tt_3 segment. The tt_0 pulse is generated by gating the $\overline{T}_1$ level against the pulse generated from T_0 going true, and the tt_2 pulse is generated by gating T_1 against T_0 going true. When placed on tape in this square wave manner, the nonlinearities

of the magnetic tape, at this frequency, cause the output, when read back, to appear as a "sine wave" rather than a square wave. The Marking program may be found in Appendix V, and the Marking flow diagram may be found in Figure 12.

STATE FLIPFLOPS

Drawing 1 lists seven of the state flipflops that form the control of the system. In addition to these there are the two motion flipflops, the motion option (i) flipflop, and the C register. The RTPRDY (Read Tape Ready) and the WRTPRDY (Write Tape Ready) essentially form one tape level, or state, which is called TPRDY (Tape Ready). The purpose of having two flipflops was to distinguish the type of data transfer in which the central machine was engaged, in case it was prone to forget. However, the testing system did not utilize this option, but merged the two to form the general TPRDY level. The Check flipflop is only used during a WRC (Write and Check) operation, since this is the only operation requiring two different phases which must be distinguished from one another. The Check flipflop is <u>not</u> used during a CHK (Check) or RDC (Read and Check) instruction. The Search flipflop defines the SEARCH phase of operation which locates the desired block on which the instruction is to be executed. It over-rides all other states in a MTP (Magnetic Tape Instruction). The Fail flipflop simply records any failures that may occur in the execution of any instruction or in the Marking process. Except for a RDC or a WRC instruction (where the system checks the contents of Fail, and if set, re-executes the instruction), the use of the information stored in the Fail flipflop is left to the discretion of the central computer. The WGFF (Write Gate Flipflop) is used only during a WRT or a MRK instruction and asserts the writing-enable levels for the Reader Writer cards on the data channels. The MarkFF enables the Mark clock, the writing levels for the Reader Writer cards attached to the Mark and Timing channels, and provides the levels necessary to execute the MARK phase of operation. The MarkFF over-rides all other states in the system.

The Motion flipflops are decoded in the following manner:

Table 2:

Code	$MOTN_1$	$MOTN_0$
BWD	1	1
Halt	0	0
BWD	0	1
FWD	1	0

The Motion flipflops control all tape motion and can only be over-ruled by the Motion buttons on the Tape Transport.

The i-flipflop allows the user a motion option after completion of the tape instruction. Use of the motion option is described in the Motion Option section.

The U register holds the unit selection number which is initially gated into the system by the MTP. This controls the tape unit and the set of tape heads which are activated by the tape system.

The C register controls the particular instruction which is executed by the machine. The three bits can be decoded into 2^3 possible states. These codes are defined as follows:

Table 3:

CODE	No.	Description
RDC	000	Read and Check desired block.
RDE	001	Read desired block.
SCH	010	Search and find desired block.

Table 3 (continued):

CHK	011	Check desired block.
WRC	100	Write and Check desired block.
WRI	101	Write desired block.
MRK	110	Mark-enable level.
MTB	111	Initiate tape motion towards indicated block. This is also the recognized Done level.

In order to avoid the use of another state flipflop, the DONE state has been hidden in the MTB instruction. After completion of any Magnetic Tape instruction, the entire C register is set to the 1 state. Clearly, $\overline{\text{DONE}}$ or $\overline{\text{MTB}}$ implies that the system is in the process of executing an instruction.

DATA REGISTERS

The S register contains the block number upon which the instruction will operate. It is gated into the A register for block number checking. If a tape contains 1000_8 blocks only 9 of the 12 bits of the S register are used.

The A register or accumulator is the register in which all 2's complement addition is done. During the SEARCH phase, the complemented block number from the tape is added to the desired block number loaded in the A register from the S register. During the operation phase, each word passing through the B register is added to the accumulator, thus forming the check-sum. All addition is done between the B and A register with the sum appearing in the A register. In addition to being loaded from the S register, it may also be cleared.

The B register acts as the data transfer window. All bits appearing on the data channels on tape flow through the B register during all instructions except WRT and MRK, in which data is placed on the data channels rather than being read off them. Thus, the B register may be thought of as three, four-bit shift registers. The data flowing into B_0, B_4, and B_8 come from the output of the Reader Writer cards, while the output flowing from B_3, B_7, and B_{11} go to the D register. Once a complete data word has been formed in the B register, it is shifted into the $\ddot{BU}$ register. Likewise, during the write process, a complete word is shifted from the $\ddot{BU}$ register into the B register for shifting onto tape.

The $\ddot{BU}$ register acts as the interface data register to the world. It holds

the data coming from the central machine until the B register is ready to absorb it, and likewise, it holds the data coming from the B register until the central machine is ready to absorb it. Due to the word-cycle-time of the system, there is a 160 microsecond limit as to the length of time the $\ddot{BU}$ register can hold a word before failure.

The WIND register (Window) is the register which continuously examines the mark track. Like the B register, it is a four-bit shift register fed by the Mark channel Reader Writer card. The output of $WIND_3$ feeds the M flip-flop of the D register which controls the Mark Reader Writer card during the Marking process. During the Mark process, the WIND register is loaded from the W register which acts in the same manner as the $\ddot{BU}$ register during the Mark Process.

The W register functions only during the MARK phase, and is responsible for interfacing the Mark channel with the central computer. During the MARK phase it is loaded the same time as the $\ddot{BU}$ register is loaded.

The T register is used only during the MARK phase, and is essentially a Mod 8 index register. Its operation is explained more thoroughly in the description of the MARK phase.

TIMING LOGIC

This system employs pulse logic which is initially generated from the Timing track of the magnetic tape. The Timing track on tape contains a "sine" wave which is amplified and clipped by the Reader Writer cards. The signal is then sent over to the main frame where the DEC logic levels are converted to MECL levels. A line driver is used as a Schmitt trigger to keep the Timing waveform as clear as possible. The resulting square wave is terminated in 100 Ω and fed into a "one shot" which creates a pulse duration of 150 ns. This pulse length was found to be sufficiently long to insure detection and sufficiently short to insure that the system would come to rest long before the following pulse. One pulse is generated by the upward transition of the square wave, and is referred to as t_2; likewise, another pulse is generated on the downward transition of the square wave and is referred to as t_0. These pulses are gated with three integrating delays before being allowed to become system timing pulses.

The first delay is called the $\overline{\text{XTLK}}$ delay. Its function is to gate out any spurious noise pulses that might appear following the true timing pulse during a WRT instruction. Because 6 millivolt signals must be detected while 36 volt (peak to peak) signals are being placed on the Data tracks of the tape, a $\overline{\text{XTLK}}$ delay (no cross-talk) is essential. Its duration is set for 10 microseconds. The tape timing pulses are 20 microseconds apart.

The second delay is called the TTOK (Tape Time OK) delay. Its function is to insure that the tape is up to the minimum speed required for reliable operation. It is set conservatively for 30 microseconds. Under normal tape conditions, it should be reset every 20 microseconds, and hence,

should never go off. Note that both the $\overline{\text{XTLK}}$ and TTOK delays are fired by the trailing edge of the t_0 and t_2 pulses. This insures that if $\overline{\text{XTLK}} \cdot \text{TTOK}$ is true, the initial pulse will get through and no other pulse will pass for at least 10 microseconds. However, if TTOK goes off, the first pulse hitting TTOK will not go through. Hence, no timing pulse will be seen unless TTOK is on continuously.

The final gating delay is the ACIP (Acceleration in Progess) delay. This delay is fired each time the tape direction is changed. Its function is to blind the tape system whenever the tape is not up to speed. However, it is different than the TTOK delay in that it is set such that during a tape turn-around at the end of a block, the system will not stop until it is well in front of the block. This speeds up tape operation considerably when one is doing many redundant tape instructions. However, it is <u>not</u> essential to the reliable functioning of the system. The timing varies slightly with the transport, but is usually operated with a delay of about 110 milliseconds.

Of the three integrating delays, only the XTLK delay is absolutely essential, and this is true only during a write instruction. The ACIP delay is there for convenience, and helps speed up consecutive tape operations. The TTOK delay is present for increased reliability, particularly in situations where the mechanical speed of the tape system may fluctuate considerably. (e.g., It would rescue the system in the case where an individual leaned against the tape unit while it was executing an instruction.)

ELECTRICAL CONSIDERATIONS

In the debugging process, it was soon found that any pulse line that was over 18 inches in length, and had more than four gates attached to it, needed to be driven by a line driver and terminated in its characteristic impedance. Loading and driving the line in this fashion had two effects. First, it cleaned up the pulse waveforms and cut down on overshoot. Second, the heavy loading cut down on cross-talk considerably. The precise characteristic impedance had to be determined empirically for each case as the characteristic impedance was not only determined by the wire used and the ground-return geometry, but also by the number and distribution of the gates placed on the line. The impedance was found to vary from 100 ohms down to 40 ohms in high gate density cases. As a rule of thumb, terminating with 100 ohms should suffice in most instances. Due to power dissipation within the MC365 line driver, 30 ohms is about the lower limit on output impedance. It was not necessary to use line drivers on any of the level lines as the lines always settled down sufficiently by the time the gating pulse arrived. Thus, it suffices to keep only the pulse line clean and terminated. The positions of the load resistors are indicated on the drawings. In cases of line branchings, the longest line is always the one terminated. However, in many cases, there is more than one termination per line.

To transform the input signals to MECL levels it was found necessary to place Schmitt triggers on all input waveforms for sufficient noise immunity. In some cases even this was not sufficient, and RC networks had to be placed on the inputs to the triggers. The Schmitt triggers had the effect of creating the correct logic levels, and also creating compatible transition swings (less than 100 ns). The design of the Schmitt trigger is shown in

Appendix III. Since the input waveforms were DEC levels (0 volts and -3 volts (logical 1)), the Schmitt trigger sufficed for a DEC to MECL buffer interface.

Transforming the MECL levels back to DEC levels required discrete component circuitry. Basically, two circuits were used, and these may be found in Appendix III. The I transistor circuit was used for level transformation where the rise and fall times were not critical, while the II transistor circuits were used when transition times were particularly critical. The only case where this was relevant was in the outputs to the Reader Writer cards where the transitions contain the information stored on tape. The circuits were designed so that no additional power supplies were needed.

Except for the basic timing circuits, the MECL series contain all the logic elements necessary to implement any system. However, to create a "one shot", it is necessary to use hybrid circuitry and in so doing, define the pulse width. In the building of a tape system, two other timing units are necessary; an integrating delay (monostable having a 100% duty cycle), and a clock. The design of these devices is found in Appendix III.

The supply used was a five ampere, 5.2 volt supply. Because of the logic transition speeds, it was necessary to use a short (36") low impedance (#18 stranded wire) power line connecting the power supply to the main frame. Within the main frame it sufficed to use a common ground return for all the logic. The -1.15 volts bias was supplied by one bias driver card containing ten MC354 bias drivers. The bias voltage was bussed around the system in the same manner as the power supply voltage and ground. The total power consumption was 15.6 watts, drawing exactly 3 amperes current.

A full description of the MECL line, and the notation used in the drawings is found in Appendix II.

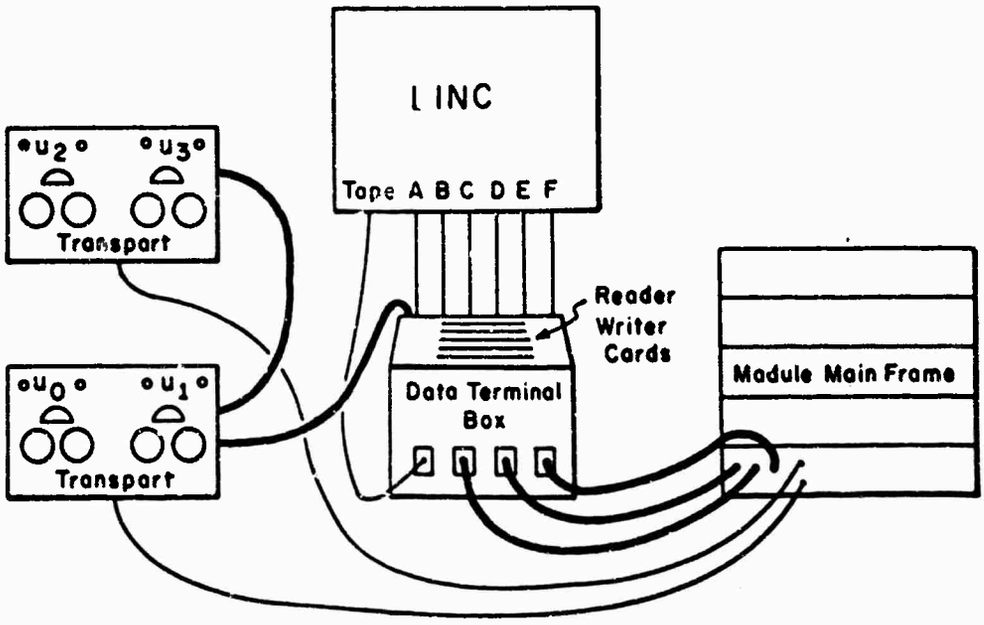
Two types of cable were used to transmit information between frames. Between the LINC and the tape system logic, three 32 connector cables were used. As the ADC connectors used to hold the system cards have 62 channels per connector, 31 on each side, two cables were connected to a single card leaving two wires floating. The third cable was connected to one side of a second card, the 32nd wire being used as a ground on the opposite side of the card. Each connector was made up of number 22 wire which has a resistance of .01614 ohms/ft. A common ground return was used. The cable length was fifteen feet between the frame and the box, while the cable length between the data terminal box and the central computer was 30 feet. As a result of the line length and the cross-talk noise, it was necessary to place softening cards on the output of the Buffered A register, and the OPR level outputs. Softening simply consists of placing a 560 ohm resistor in series with the output. Where this was not sufficient to quiet the levels, .01 ufd. capacitors were added to filter out the high frequency noise. This was done on the external level inputs, and on the MTP, MARK, and Preset inputs to the tape system.

The second type of cable used was the shielded twisted pair, 32 connector, 16 channel. This was used to carry the 6 millivolt signals from the tape head to the Reader Writer cards. This cable was used only between the tape transport and the data terminal box. No problems were encountered with its use, even when 45 feet of cable were used to connect the two transports.

TESTING PROCEDURE

This system was tested using the LINC (Laboratory Instrument Computer) as the central processing module. In the LINC, all the necessary logic levels are brought out to a "data terminal box". Likewise, the input and output cables of the tape system module were connected to the terminal box, and all necessary interconnections between the two systems were made there. Because all the required voltages to drive the Reader Writer cards are available in the data terminal box, the cards were placed here. Thus, the terminal box also acted as the interface between the tape transport and the main frame of the tape system. See Figure 10.

Figure 10.



The data were transmitted by means of the input and output data lines of the LINC; the TN sense lines, and the gated A and Buffered R output lines. This made it possible to use the R register to hold 1) the unit number and i-bit option, and 2) the Mark bits during the Marking operation. Likewise, it was possible to use the Accumulator as the input-output window for all instructions and data. Thus, when the initial MTP is sent over to the tape system, the Accumulator holds the block number and instruction code, while the R register holds the unit number and i-bit option. When the reading or writing process begins, all data flows through the accumulator.

The generalized flow diagram of the required operation of the central computer is shown in Figure 11. In the LINC, the first twelve bits of the R1 register are the A register or accumulator. The last three bits of R1, which contain the unit number and the Motion Option, are the first three bits of the R register (R_0, R_1, R_2). R2 is formed by the first four bits of the R register (R_0-R_3). The data register, R3, is the accumulator. The S register, or memory address register referred to, would simply be the S register in the LINC. Provision has been made for a 12 bit, or $10{,}000_8$ block numbers, in which case the R1 register would have to be extended to 18 bits. However, experience on the LINC has shown that 1000_8, 400_8 word blocks are a convenient tape format, hence, only nine bit block numbers are presently utilized.

The Marking routine, which in addition to a hardware interface, requires a software program to generate the correct marks and block numbers, is shown in Figure 12. The general software system is shown in Figure 13. The actual programs used on the LINC are shown documented in Appendix V.

MAGNETIC TAPE INSTRUCTION

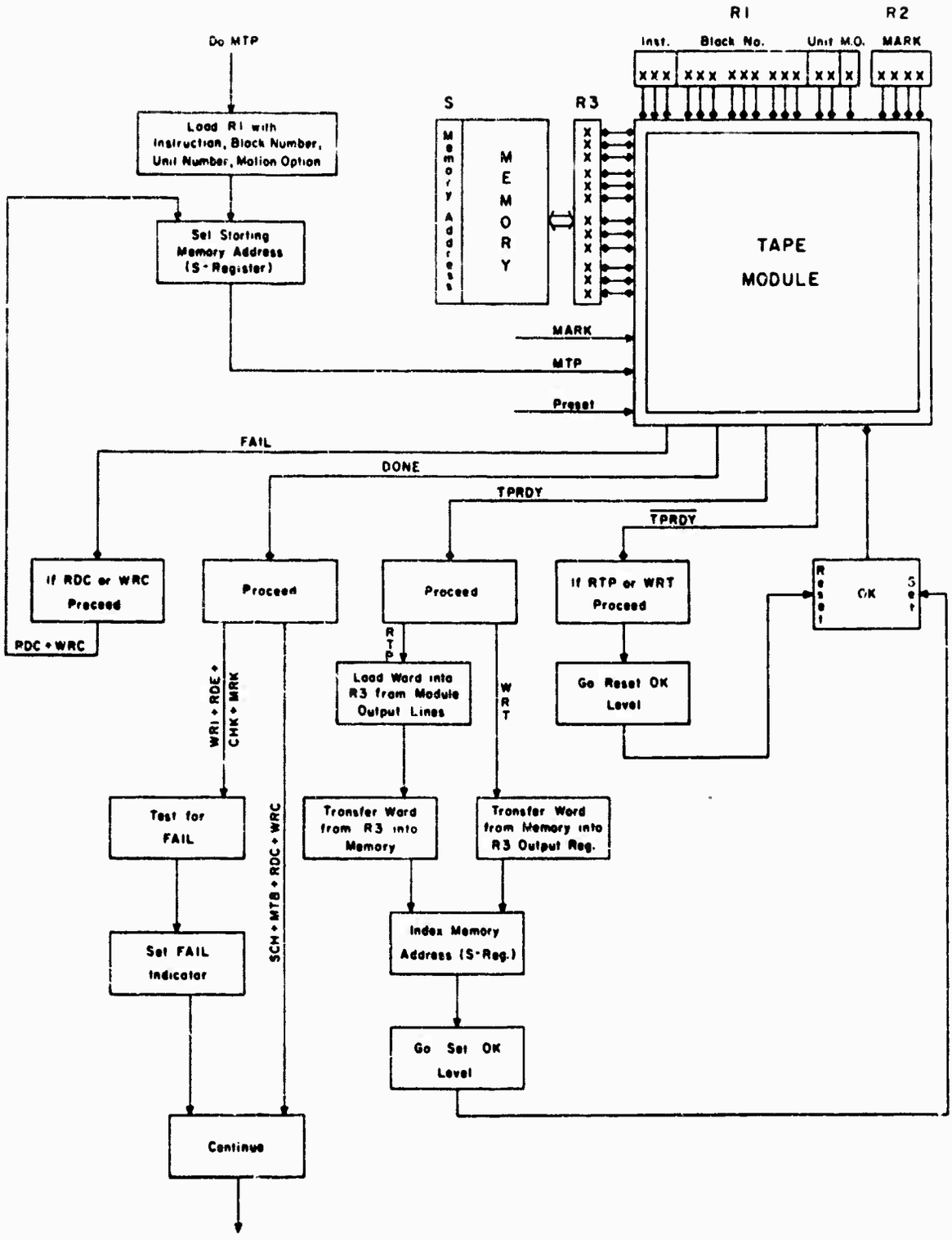


Figure 11

MARKING SYSTEM

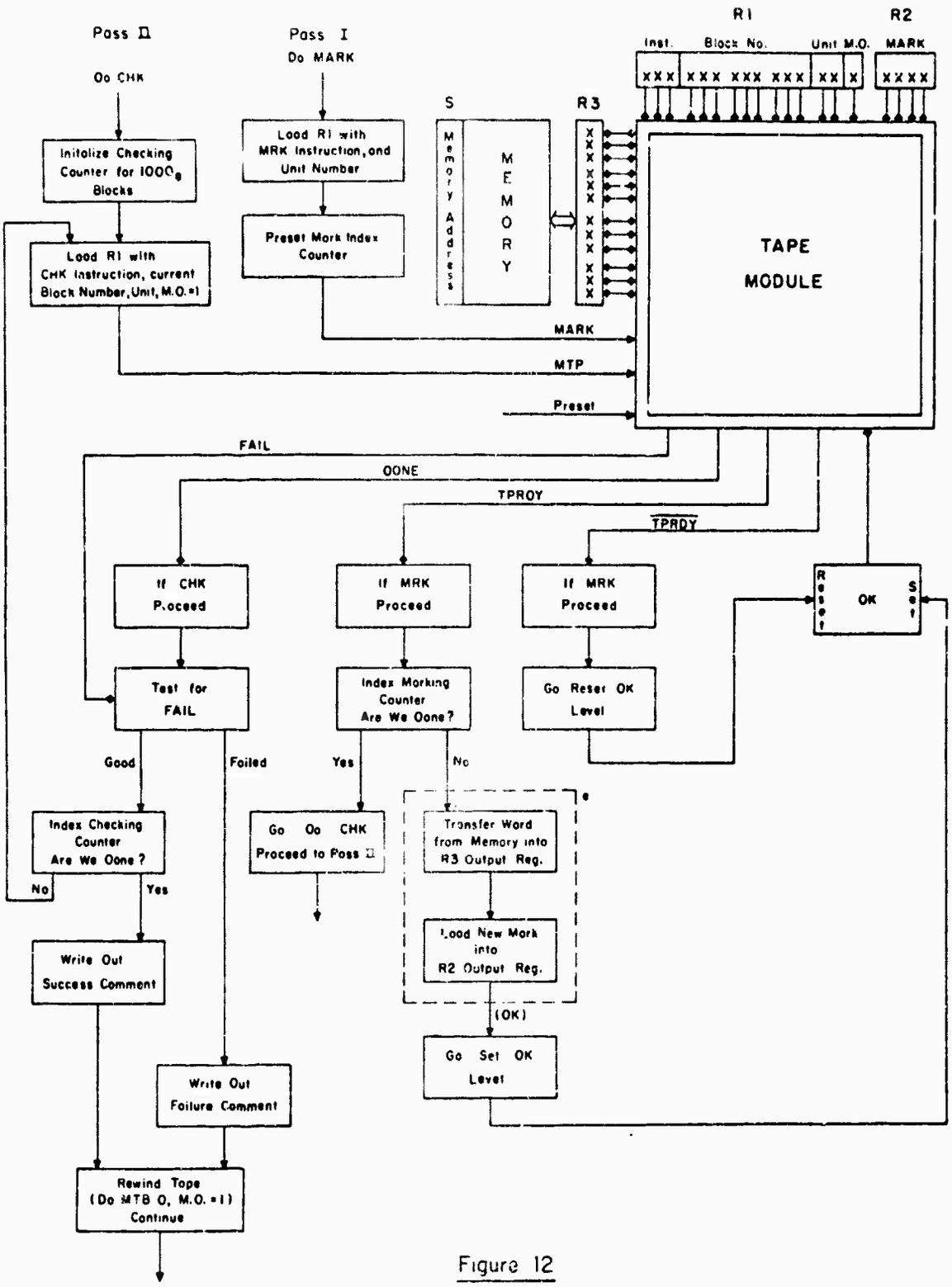


Figure 12

*See Fig. 13 for more detailed Flow Chart.

SOFTWARE MARKING ROUTINE

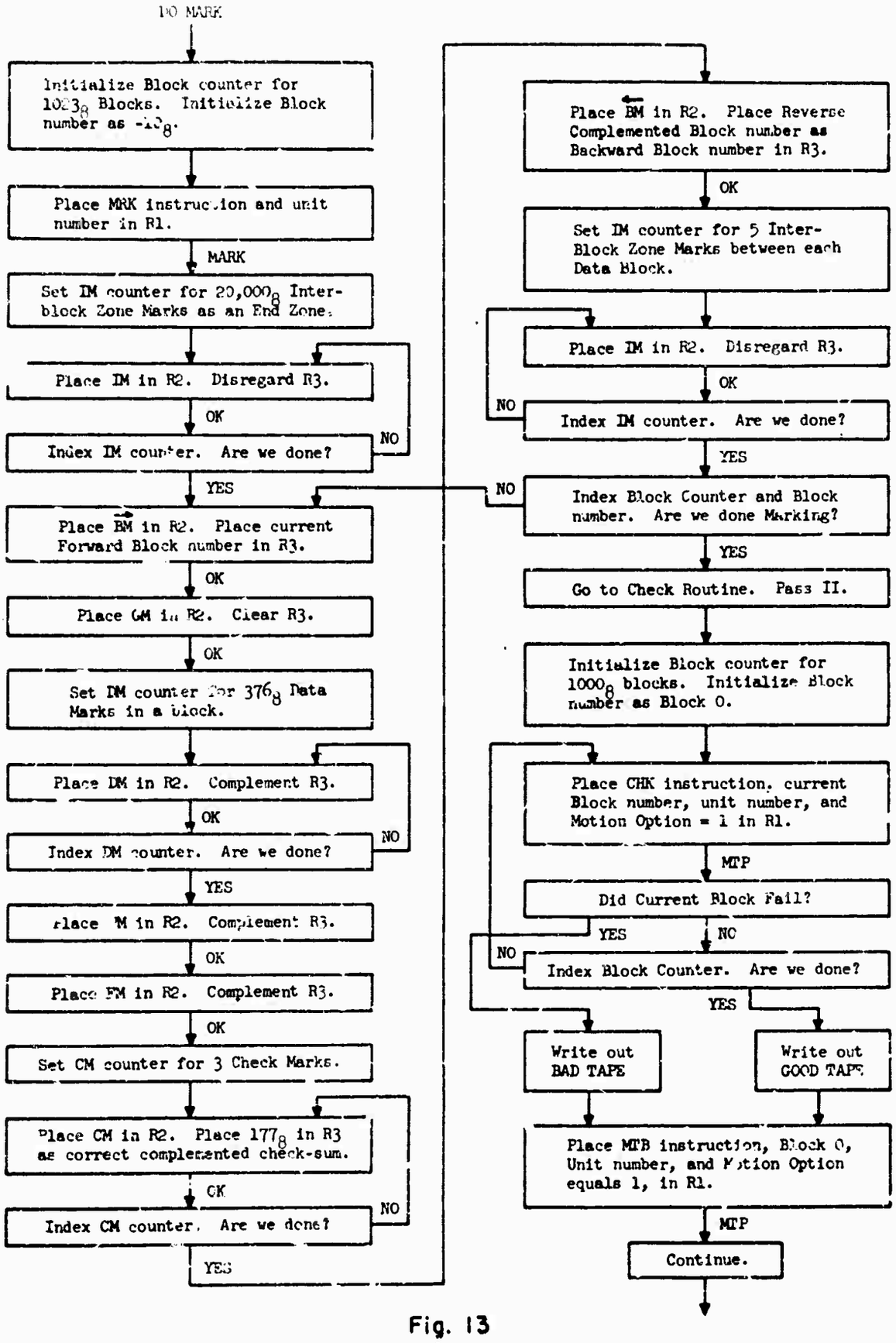


Fig. 13

The arrows shown entering the Tape Module are control pulses, but need not be mor than negative transitions covering the MECL voltage range. The diamonds going to and leaving the module indicate levels which remain on until the condition is no longer true. The diamonds entering from R1 and R2 are simply levels that must be correct when the MTP is asserted. After that time, the information in R1 and R2 may be changed as the module stores the information initially. The R3 data register has diamonds going both ways indicating that information can flow either way depending upon whether it is executing a RTP or a WRT instruction. During a RTP instruction, the TPRDY level indicates that the information is on the data lines and is ready to be loaded into R3. The return OK indicates that the information has been recieved and that the data lines may now be altered. During a WRT instruction, the OK level indicates that the information in R3 is correct, and will remain so until the $\overline{\text{TPRDY}}$ level appears.

After the MTP has started the Tape Module, the central computer is assumed to be either paused or operating on something else until one of the four control levels appears and starts the central machine worrying about tape business again. During a MTP instruction, the Fail and $\overline{\text{TPRDY}}$ levels are conditional, depending on the type of instruction being executed. The only instructions in which a Failure will change the operation (cause the execution to be repeated) are a RDC and a WRC. The only instructions affected by a $\overline{\text{TPRDY}}$ following a TPRDY are information transfer instructions (RTP and WRT). The TPRDY level is not considered conditional since the only Magnetic Tape instructions in which it can appear are RTP and WRT instructions. The Done level is nonconditional and terminates all instructions. The post-Done Failure test simply amounts to sampling the FAIL line coming from the Module.

If there has been a Failure, it will be indicated until the next SEARCH phase has been completed, or until the system has been Preset.

The arrows going to the box marked OK, indicate the setting and resetting of the OK level or flipflop. Upon execution of this step, the central machine goes back into the Pause state, or returns to the business with which it was previously occupied, and waits for the Module to respond again on one of the control lines. It should be noted that if pulses are used instead of levels for MARK, MTP, and Preset, they should be of at least 20 nanoseconds duration. There is, however, no upper restriction on the length or risetime of the pulse or level.

The Marking System operates in much the same way except that during the actual Mark phase, only the TPRDY and $\overline{\text{TPRDY}}$ levels are operated upon. The Mark phase is not terminated until the central machine executes a MTP instruction. This immediately takes it out of the Mark phase and places it in the Search phase of a MTP instruction. Clearing of the MarkFF places the MOTN flipflops in reverse. Pass II merely consists of consecutive checking of blocks. The FAIL level, in this case, is merely sampled after the DONE level has appeared, but does not itself initiate central machine operation.

The control pulses and levels are all handled by means of the OPR (Operate) instructions. This instruction makes a level available for as long as the instruction is being executed. Basically, there are two instruction options; the first merely asserts a level for 16 microseconds, and then continues with the next instruction: the second, (an OPR i) holds the level on until

a condition is met from the outside world. Both the OPR and OPR i instructions are used. In addition, an SXL (Sense External Level) instruction is also used so that several conditions may be sampled without "hanging up" while waiting for just one condition to be met.

The three basic pulses that are recognized by the tape system are the MARK pulse, the MTP (Magnetic Tape Pulse) and the Preset pulse. The Preset pulse is supplied by the LINC through the buffered Preset output line. Hence, if the power is initially on, or comes on with the LINC power, then the entire system is preset. Hitting the Stop key on the LINC console also Presets the system. The MARK pulse is generated by an OPR 7 instruction which applies a 16 microsecond pulse to the OPR 7 output line. Likewise, the MTP is generated by an OPR 3 instruction. All these pulses are shaped by a Schmitt trigger on arrival, and in turn fire one-shots yielding a set 150 ns pulse.

The TPRDY (Tape Ready) output is fed into the LINC on sense line 4. Likewise, the $\overline{\text{TPRDY}}$ output is fed in on sense line 5. The OPR i 5 output line is interpreted by the system as the OK level. Hence, the OK level is immediately terminated by the LINC upon receipt of the $\overline{\text{TPRDY}}$ level, since this is by definition the condition which satisfies the OPR i 5 instruction. When $\overline{\text{TPRDY}}$ is received, the LINC proceeds to the next instruction. The purpose of an OPR i 4 instruction is to pause the LINC until the TPRDY level comes up and to load the Accumulator with the data on the input lines. Sense line 6 is used as the DONE level input to the LINC, and sense line 2 is used as the FAIL flag input to the LINC. An SXL 6, or an SXL 2 instruction, will sample these lines to determine whether these conditions do indeed exist.

It should be noted that the block structure used by this system is precedented by the LINC system. One thousand octal, 400_8 word blocks has proven to be a very useful format, and there was no justifiable reason for changing it in the initial system. The choice of five IM's to form the inter-block zone was simply borrowed from Ornstein's Mark program[4]. Likewise, creating 10_8 negative blocks before the useful block area, and 13_8 blocks beyond the useful block area was also a feature adopted from the Ornstein routine. This creates a buffer zone of sufficient latitude that should the system overrun the limits of the useful block area in normal SEARCH operation, it will always be returned to the central zone by the buffer blocks. Again, these lengths were chosen since they had proven themselves in constant system usage. As the initial goal was to make the system operational, it was desired to keep the number of variables as small as possible. Hence, block lengths, inter-block zone lengths, buffer-block zone lengths, and end zone lengths were chosen initially, as system-proved constants.

The original tapes used in the debugging of this system were marked on the LINC. This was done by taking the Ornstein routine and modifying it so that the new codes replaced the old codes. Fortunately, no errors were made in the initial marking process, hence, removing one variable from the debugging headache. After the Marking system had been debugged, the module itself was used to Mark tapes. The software used to Mark in conjunction with the LINC is shown in Appendix V.

---

[4]Severo Ornstein and Charles Molnar, "LINC Theory of Operations II", (St. Louis: Washington University, 1967), XIV.

Due to the LINC Marking system, there is a polarity inversion inco rated on the Timing track. Note that $\bar{T}_1$ levels, rather than T_1 levels are employed in the Marking system. This inversion can be corrected if both the input to the Timing track Reader Writer card, and its output are inverted. This was not done in the present system, however, since it was desirable to be able to use the LINC Marking system to initially Mark the tapes. Correcting this logical inconsistency would negate the LINC's utility in Marking tapes.

Provision has been made for operation of more than one tape transport. In this situation, the unit selection line is an independent cable going from the main frame of the tape system to each of the tape transports individually. Depending on which line goes to which transport, each of the tape units will be designated by a different octal number, with the even number always designating the left tape unit, and the next succeeding odd number designating the right tape unit (see Figure 10). The present tape system accomodates only two tape transports, but is easily expandable to eight or more transports. At any one time, only one tape unit can be activated. Hence, the addition of more tape transports will not necessarily speed up tape operation, it will simply give you access to more tapes.

To accomodate this change, the jumper plug in the back of the tape transport must be modified such that it becomes the male unit selecting input plug. The change is made by cutting the connection between pins A and B, and between pins C and D. The right unit selection input wire is connected to pin A, and the left unit selection input wire is connected to pin B.

When only one tape transport is being used, no separate unit selection line

is needed, as that information is supplied by the shielded data cable. In this case, the unmodified jumper plug is used which has the pins paired off as follows: A to B, C to D, E to F, and H to J. For more detailed information on the operation of the tape transport, see Volume 5 of the LINC drawings[5], and Volume 14 of the LINC Theory of Operations, II[6].

---

[5]Wesley Clark and Charles Molnar, "The LINC", Proceedings, New York Academy of Sciences Conference on Computers in Medicine and Biology, May 27-29, 1963.

[6]Severo Ornstein and Charles Molnar, "LINC Theory of Operations II", (St. Louis: Washington University, 1967), XIV.

ECONOMICS

The cost breakdown of the entire tape system is presented in Table 4.

Table 4.

One-hundred card Main Frame (Scande Corporation)	$ 582.90
Case surrounding Main Frame (Scientific Atlanta)	161.00
65 cards ($11.00 ea.), (Applied Development Corp.)	715.00
Electronics	1425.00
Reader Writer cards (5 at $200.00 ea.)	1000.00
Wire	100.00
Cables	20.00
Tape Transport	3100.00
Total system cost:	$7108.90

The interesting point to notice is that except for the Reader Writer Cards, the Tape Transport costs more than the rest of the system combined. Note also that the electronics cost less than the trappings needed to hold it. It should be noted that no labor costs for assembly have been included. A rough estimate of assembly time would be about one good technician week. Likewise, one engineer week should be sufficient to debug it. Thus, total assembled system cost should run about $7500. As these costs have been computed on a small quantity basis, further saving might accrue through buying in quantity. The Reader Writer cards absorb a disproportionate fraction of the cost as the actual electronics contained on the card cost less than $50. Using integrated circuits instead of DEC System Modules, one might well build their own for about $50.00, today.

Thus, for $7500.00, one is buying himself an entirely self-contained, asynchronous tape system with a bit rate of 75 kiloHertz. As the module needs only one transfer every 160 microseconds, the central machine is liberated for the remainder of the cycle time to take care of other business. If the tape system is at rest, the minimal amount of time necessary for one block transfer would be 150 ms. If, however, the system is in the forward direction, then the minimal time would be reduced to 40 ms. Thus, one achieves a quite respectable data rate at a rather conservative price.

APPENDIX I: INSTRUCTION CODE

A. Read Instruction (RTP)

001 RDE: The Read operation searches for the indicated block and initiates transferal on the first Data Mark by turning on the Read Tape Ready level. The central computer has 160 microseconds to absorb each word after receipt of the level. Absorption is indicated by the raising of an "OK" flag. When the flag is sensed by the tape module, the Tape Ready level is turned off. The cycle is completed when the central computer turns off its OK level. Data transferal is terminated with the sensing of a Check Mark. The module then compares the check-sum it has been forming throughout the transferal with the one recorded on tape beneath the Check Mark. If they are not equal the Fail level is turned on. In any case, the DONE level is turned on and the Motion Option is executed.

000 RDC: The Read and Check operation is identical to the Read operation except that termination of the instruction is conditional. The DONE level is not raised until the two check-sums are equivalent. If a Failure occurs, the module is placed back into the Search phase and the instruction is re-executed. Upon successful completion of the instruction, the DONE level is raised, and the Motion Option is executed.

B. Write Instruction (WRT)

101 WRI: The Write operation searches for the indicated block, and initiates transferal on the sensing of the desired block number, by turning on the Write Tape Ready level. The central computer has 160 microseconds to place each word in the window before it must be absorbed by the tape module. When the word has been placed in the window, the central computer indicates by raising an "OK" flag. Absorption of the data word is indicated by the turning off of the WRTPRDY level. The cycle is completed when the central computer turns off the OK level. Transferal is terminated with the sensing of the Prefinal Mark, and the WRTPRDY level remains off. The module then records the complemented check-sum

it has formed during transferal underneath the Check Mark. The DONE level is subsequently turned on and the Motion Option is executed.

100 WRC: The Write and Check operation is identical to the WRI operation except that termination is conditional upon a successful Check pass. After the check-sum has been recorded, the module goes into the Search phase and executes a Check. If the new check-sum is not equivalent with the one stored on tape, the entire instruction is re-executed. Upon successful completion of the instruction, the DONE level is raised, and the Motion Option is executed.

C. Non-transferal Instructions

011 CHK: The Check operation searches for the indicated block, and forms a two's complement check-sum of the data words in that block. This check-sum is compared to the one recorded underneath the Check Mark. If it is not equivalent, the Fail flag is raised. In any case, the DONE level is turned on and the Motion Option is executed.

010 SCH: The Search operation searches for the indicated block. Upon successful location of that block, the DONE level is turned on and the Motion Option is executed.

110 MRK: This is simply the Mark enabling code, without which the machine is unable to Mark. Should it be executed as a MTP instruction, its execution will be identical to that of the SCH instruction with the exception that a Motion Option of 1 will always leave the tape going in the forward direction.

111 MTB: The Move Toward Block operation reads the first Block number it comes to and then: 1) reverses tape motion if the corresponding Block number is larger than the one it is trying to move towards; 2) continues in the same direction if the corresponding

block number is smaller than the one it is trying to move towards. The motion option is then executed. Since the MTB code forms the DONE level, the instruction is considered executed and completed as soon as the code has been received by the Tape Module.

APPENDIX II: LOGIC CONVENTIONS

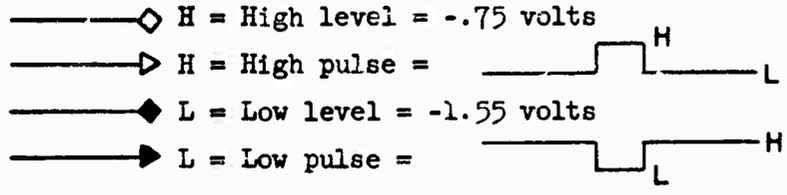
A. 350 Series

Rather than choosing any polarity to be a logical "1", or a logical "0", we are prone to think of gates performing logical functions yielding a "True" or "False" state in a particular "sex" (polarity) as defined by that logical unit. Thus, we can assert with either sex depending, of course, on just what logical operation is desired of a particular gate. In as much as we have been seduced by Motorola's MECL (Microelectronic emitter-coupled logic), the proceeding conventions will be used for the components available in the MECL 350 series. An exception will be made for the MC354, since it is not a logical unit but a bias supply.

It is necessary to have one load resistor (2K) on all gate outputs. If one uses emitter gating between logic gates, then it is desirable to have only one load resistor on the emitter-tied line. All outputs having a built-in load resistor are indicated. The resulting gate fanout is about 15. Circles indicate inverting outputs.

Voltage and Polarity Conventions:

pin 3 V_{CC}=0 volts
pin 2 V_{EE}=-5.2 volts
pin 1* V_{BB}=-1.15 volts



For all gating, to achieve a Logical "OR", one uses High for assertion. For a Logical "AND", one must use Low for assertion. (To "OR", define H = Logical 1; To "AND," define L = Logical 1.)

*352, 355, 358, 362A, 364 excepted!

In the case of tied-emitter gating, the High voltage always wins, therefore, the same rule applies: High asserts to "OR", Low asserts to "AND".

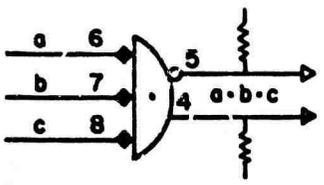
In as much as all the gates are basically "NOR" (or NAND depending on definition of Logical 1) gates in the classical sense where absolute logical values are assigned respectively to each voltage, the logical operation sign will be noted in the middle of the gate without respect to output sex. Hence, gates will be called "AND", "OR", or "NOT", for ease in Boolean decoding.

B. Three Input, Two Output Gate

Options: 1) MC 356: Load resistors on both sex outputs.
2) MC 357: No load resistor on either sex output.
3) MC 365: No load resistor on either sex output.

AND Operation: Define low inputs as Logical 1, both sexes are available at out.

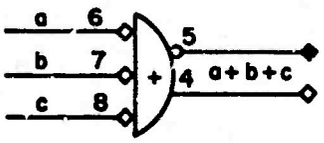
356:



a	b	c	Pin 4	Pin 5
H	H	H	H	L
H	H	L	H	L
H	L	H	H	L
H	L	L	H	L
L	H	H	H	L
L	H	L	H	L
L	L	H	H	L
L	L	L	L	H

OR Operation: Define high inputs as Logical 1, both sexes are available at out.

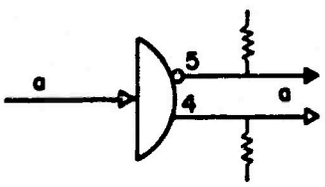
357 or 365:



a	b	c	Pin 4	Pin 5
L	L	L	L	H
L	L	H	H	L
L	H	L	H	L
L	H	H	H	L
H	L	L	H	L
H	L	H	H	L
H	H	L	H	L
H	H	H	H	L

ISOLATOR Operation: Define as you please. This is not a logical operation usage, but rather an amplification and/or isolation usage. This has the advantage over the 359 of yielding same sex out as in.

356:



C. Dual Two Input Nor

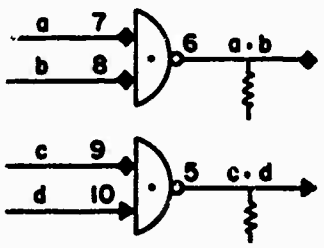
Options: 1) MC 359: Load resistors on both output emitters (both pin 5 and pin 6).

2) MC 360: Load resistor on only one output emitter (pin 6).

3) MC 361: No load resistors on either output emitter.

AND Operation: Define low inputs and high outputs as Logical 1.

359:

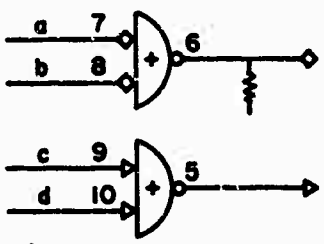


a	b	a·b
H	H	L
H	L	L
L	H	L
L	L	H

If emitters are tied together, the resulting output will be: a·b + c·d.

OR Operation: Define high inputs and low outputs as Logical 1.

360:

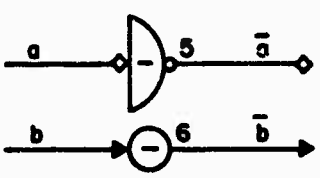


a	b	a + b
L	L	H
L	H	L
H	L	L
H	H	L

If emitters are tied together, the resulting output will be: (a+b)·(c+d).

NOT* Operation: Define as you please.

361:

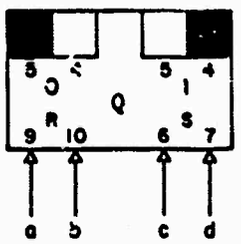


a	$\bar{a}$
L	H
H	L

*Both models are now in use. Either input may be used ("a" case: 7 or 8, "b" case: 9 or 10). Tying down the remaining input is optional, but if done, it must be to low only.

D. R-S Flip Flop (MC 352)

Operation: Define high pulse input as Logical 1. Both sexes available at output. Note that we have two Set and two Reset inputs, which are "OR'ed" together internally but isolated from each other externally.



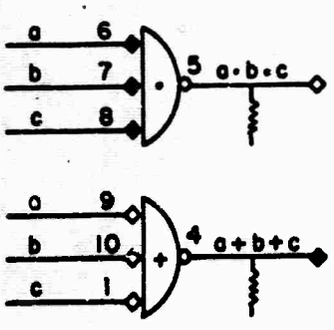
a	b	c	d	Q
0	0	0	0	q
0	0	-	1	1
0	0	1	-	1
-	1	0	0	0
1	-	0	0	0
1	-	1	-	N.D.

(let: - = 0 or 1; N.D. = Not Defined)

NOTE: THIS IS A LEVEL SENSITIVE DEVICE

E. Dual Three Input Nor (MC 362A)

Operation: Precisely the same as the MC 359 (Dual Two Input Nor) except that it is a three input gate rather than a two input gate. Due to the fact that this unit always has emitter resistors, emitter gating with other 362's is severly limited.

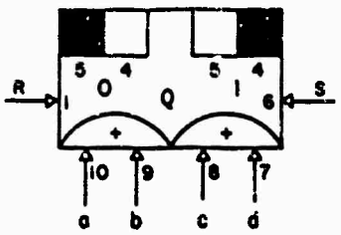


a	b	c	a·b·c
H	H	H	L
H	H	L	L
H	L	H	L
H	L	L	L
L	H	H	L
L	H	L	L
L	L	H	L
L	L	L	H

d	e	f	d+e+f
L	L	L	H
L	L	H	L
L	H	L	L
L	H	H	L
H	L	L	L
H	L	H	L
H	H	L	L
H	H	H	L

F. J-K Flip Flop (MC 358 or MC 364)

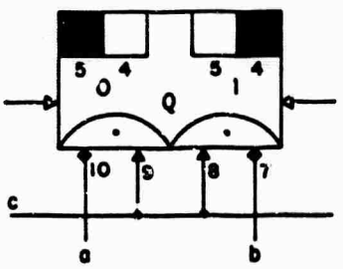
Normal J-K Operation. Define Logical 1 as an asserting pulse (transition from low to high) on input. Both sexes of levels available at output. K refers to either a or b; J refers to either c or d.



S	R	Q
0	0	q
0	1	0
1	0	1
1	1	N.D.

J	K	Q
0	0	q
0	1	0
1	0	1
1	1	$\bar{q}$

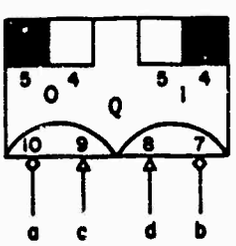
AND Gating: Define Logical 1 as low input level for a and b, and as low pulse for c. Used in this manner with negative pulses yields "Trailing Edge Triggering" logic.

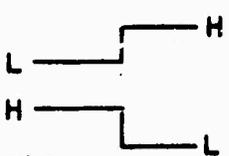


(J) (K)

b	a	c	Q
-	-	0	q
H	H	1	q
H	L	1	0
L	H	1	1
L	L	1	$\bar{q}$

As You Like It Operation:



Let U = 

Let D =

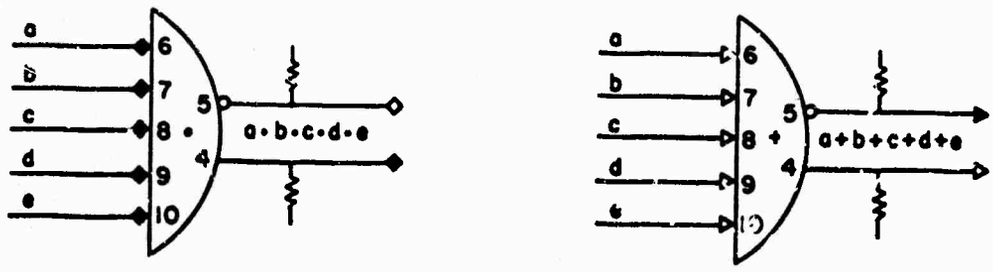
Let - = anything

Let * = anthing but U

a	b	c	d	Q
H	H	-	-	q
*	*	*	L	q
*	*	L	*	q
H	L	-	U	1
L	H	U	-	0
L	L	U	*	0
L	L	*	U	1
L	L	U	U	$\bar{q}$

G. Five Input Two Output Gate (MC351)

Operation: Precisely the same as the MC356 (Three input two output gate) except that five inputs are gated with both sexes available instead of three. Apply the same rules as you would to the MC356. Remember, since the 351 always has emitter resistors, emitter gating with other 351's is limited.

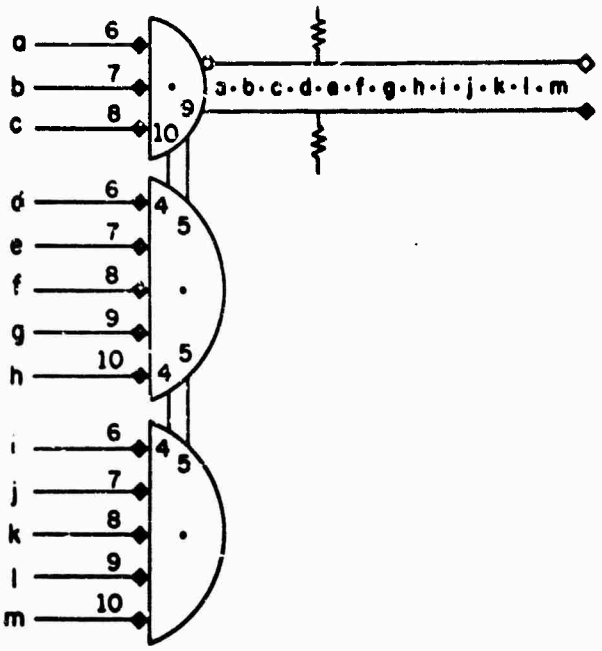


H. Five Input Gate Expander (MC355)

Operation: This is to be used in conjunction with the MC356 (or MC 357), and the MC352. It merely expands the input capacity of one basic function. Thus, apply the same rules as you would for the MC356 (or 352). The following are: 1) An example of a 13 input "AND" gate made up of two MC355's and a MC356; 2) An example of a 2 input set, 7 input reset; and, 3) A 7 input set, 2 input reset flipflop. (See next page).

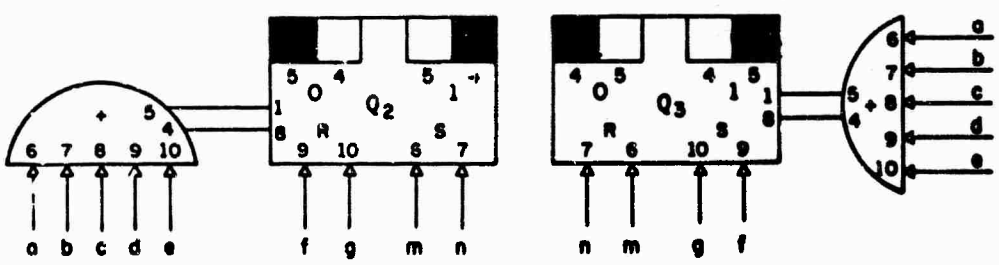
Note that the only difference between Q_2 and Q_3 is a redefinition of set and reset inputs, and "1" and "0" outputs.

1.



2.

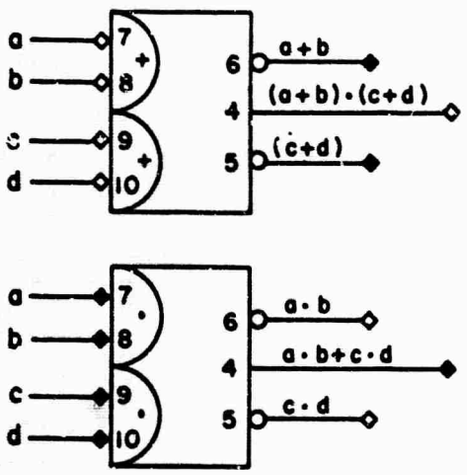
3.



I. Half-adder (MC353)

This unit is an attempt by Motorola to create in one building block a group of fairly useful functions that may occur frequently. This unit does not necessarily reduce the total price, as it is more expensive than three two-input gates; however, used judiciously, it may well cut down on the total number of cans necessary to perform a function. It can be considered only a Half-adder (in the true sense) when used in conjunction with the MC358 J-K Flipflop.

Generalized Operation: Define High as Logical 1 for inputs.



A	B	C	D	Pin 4	Pin 6	Pin 5
L	L	L	L	SL	H	H
L	L	L	H	L	H	L
L	L	H	L	L	H	L
L	L	H	H	L	H	L
L	H	L	L	L	L	H
L	H	L	H	H	L	L
L	H	H	L	H	L	L
L	H	H	H	H	L	L
H	L	L	L	L	L	H
H	L	L	H	H	L	L
H	L	H	L	H	L	L
H	L	H	H	H	L	L
H	H	L	L	L	L	H
H	H	L	H	H	L	L
H	H	H	L	H	L	L
H	H	H	H	H	L	L

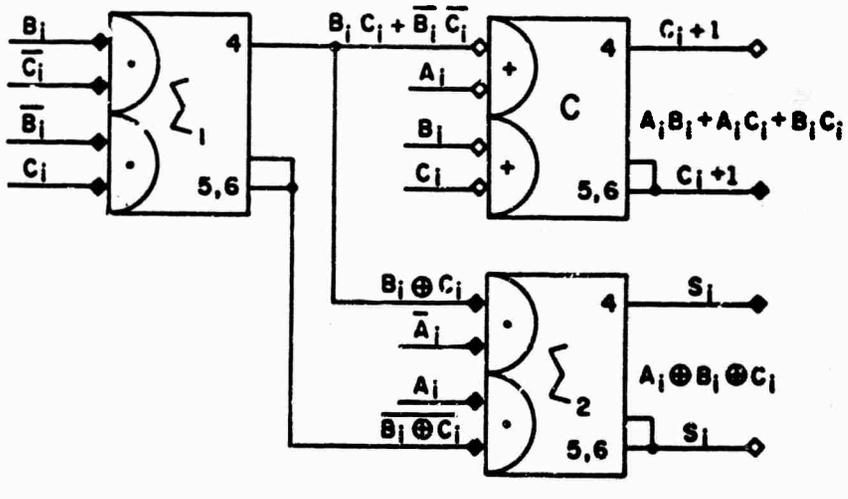
Define SL = Super Low = -2.3 volts.

Note: There is one spurious state when all inputs are low, yielding a super low input on pin 4. This state may be considered a normal low output state for all logical applications considered in this write-up. However, since super low is caused by a saturated V_{BB} transistor, recovery times may differ from those in the specification sheet.

J. Full-adder Configuration

In this application connect pin 6 to pin 5, thus yielding the complemented output of pin 4 at the resulting terminal (i.e., both sexes of the same function are now available). If one is adding to an operand in a J-K flipflop, then the Σ_2 sum is not necessary, and one merely gates the low $B_i \oplus C_i$ with an Add pulse into the J-K inputs. Define Logical one as indicated.

A_i	B_i	C_i	$B_i \oplus C_i$	S_i	C_{i+1}
H	H	H	H	H	H
H	H	L	L	L	H
H	L	H	L	L	H
H	L	L	H	H	L
L	H	H	H	L	H
L	H	L	L	H	L
L	L	H	L	H	L
L	L	L	H	L	L

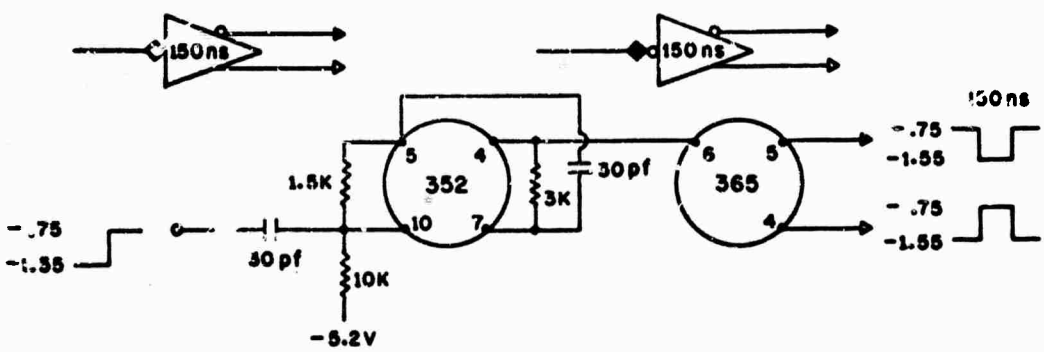


APPENDIX III: DESIGN OF TIMING CIRCUITS

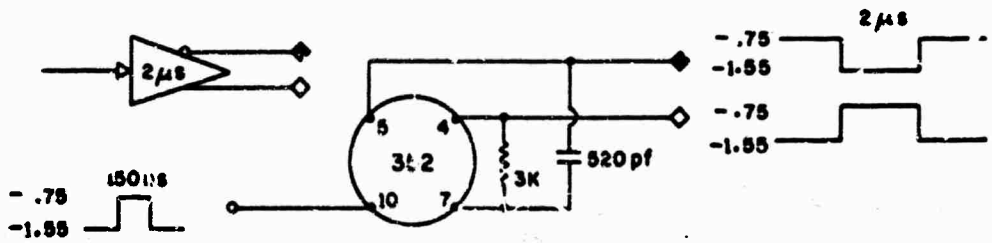
A. One Shot Pulser and Delay.

The design of the "One Shot" was taken from the Motorola application notice 233. The pulser was designed such that the RC time constant gave a pulse length of about 150 ns. The input network was used to insure that only the positive transition would fire the monostable. Those pulsers used in the system having negative transition inputs utilized one stage of a 359 inverter before being fed into the one shot circuitry. The line driver on the output was used to insure that there would be little pulse deterioration due to loading. These lines should be terminated with 100 ohm resistors for impedance matching and crosstalk quieting. The two microsecond delay differs in three respects from the pulser: the RC time constant is set for 2 microseconds; there is no input network used since it is triggered by a pulse of shorter duration than the delay; and there is no need for a line driver as the fan-out and line length were kept small.

One Shot Pulser:



One Shot Delay:

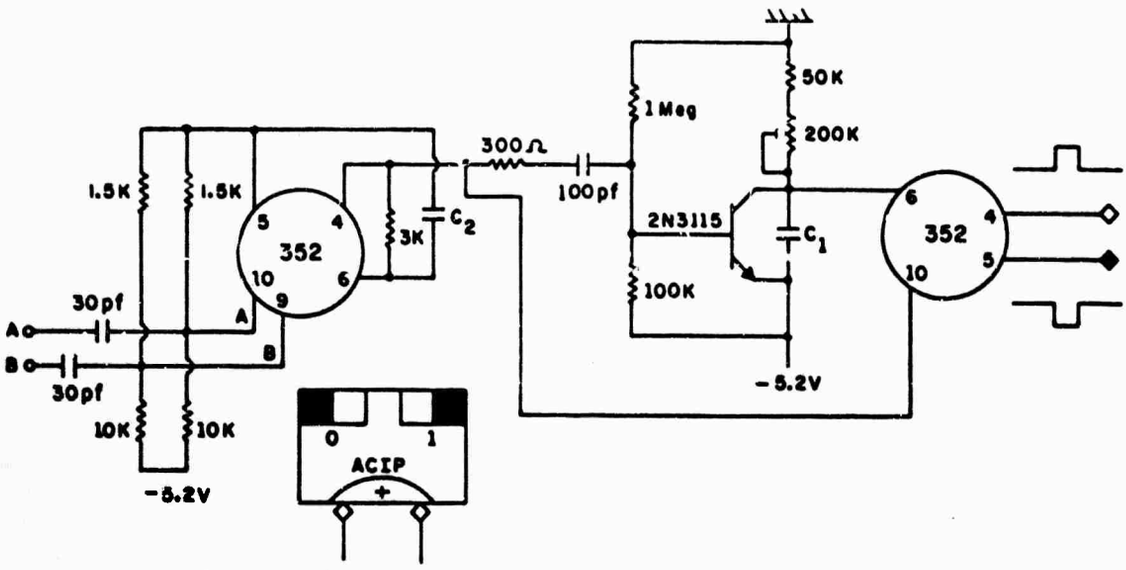


B. Integrating Delays.

Most of the development work on the integrating monostable was done at the Computer Research Labs by Thomas Chaney. The input network of the monostable was modified for the tape system's particular needs. Basically, the monostables used fall into two categories. First, the negative asserting AND gating configuration; and second, the positive transition OR gating configuration. Clearly, for single input use, the operation of the two is identical. Both circuits have the advantage of being trailing-edge triggered devices when using negative going pulses. This feature was important in that the output of the XTLK delay was used to gate the input pulse. Used in this manner, only the first pulse manages to get through, while any others that arrive during the "on" time of the monostable are gated out. Thus, both the TTOK and the XTLK delays were of the AND class monostables. The ACIP delay, on the other hand, is of the OR class in that it is desirable to trigger the monostable if either the $MOTN_0$ or $MOTN_1$ flipflops turn on (see Drawing 3).

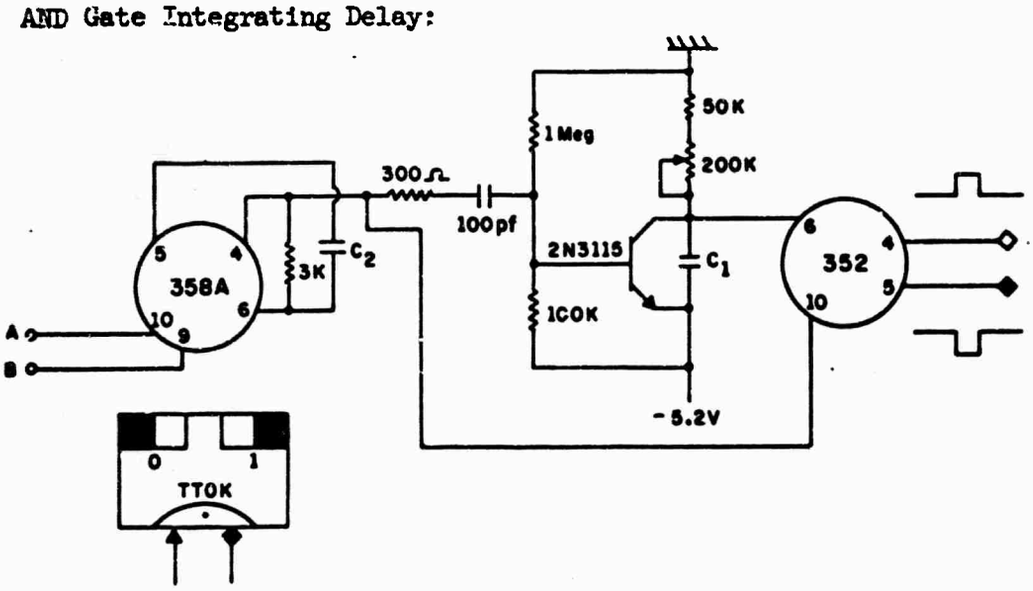
The purpose of the first stage of the monostable is to create a well-defined pulse from the input transition. This pulse is used to set the output flipflop, and discharge the reset capacitor. Since the 2N3115 transistors have a finite Beta (about 100), it takes a finite amount of time to discharge the capacitor, hence the need for a pulse of fixed duration. The pulse width determines the maximum size of the output capacitor that can be discharged on the output of the 2N3115. Using our 150 ns pulse, the maximum C_1 is about 2000 pf, yielding a maximum pulse length on the output of about 600 microseconds. Since the XTLK and TTOK delays are far shorter than this, the 150 ns RC input one shot will suffice. However, for the 110 millisecond ACIP delay, it was necessary to produce a 25 microsecond one shot pulse to charge a C_1 of .47 microfarads. After the one shot pulse has elapsed, the delay runs out as the capacitor charges. When the capacitor voltage reaches the setting threshold, the output flipflop is turned off, and the monostable has returned to its initial stable state.

1. OR Gate Integrating Delay:



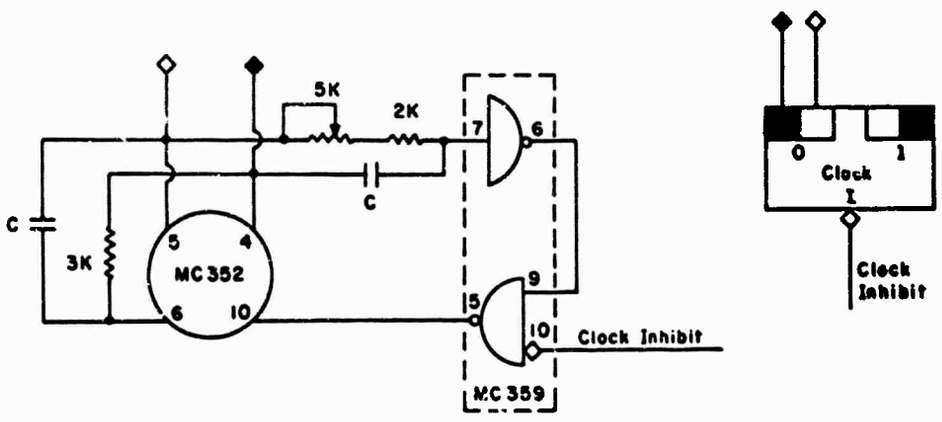
ACIP:	C_2 = .01 μfd	C_1 = .47 μfd	T = 110 ms	OR gate
TTOK:	C_2 = 30 pf	C_1 = 82 pf	T = 30 μs	AND gate
XTLK:	C_2 = 30 pf	C_1 = 51 pf	T = 12 μs	AND gate

2. AND Gate Integrating Delay:



C. Variable Clock.

The clock used in the tape system is a simple bistable device developed by Thomas Chaney at Washington University. The bistable employs an RS flipflop, and a dual-input, Nor, MEC circuit. The hybrid elements consist of resistors connecting the output of one side of the flipflop with the input of the other side; and capacitors connecting the output of one side with its input. When the capacitor connected to the "on" (High) side of the flipflop becomes sufficiently discharged (reaches about -1.15v.), that this side of the flipflop is no longer setting itself, the opposite side is free to set itself through the resistor connected to the "on" output. This causes the flipflop to switch states, thus, totally discharging the first capacitor, and causing the second capacitor to fully charge. This holds the flipflop in its new state until that capacitor discharges. The purpose of the dual input Nor is simply to allow the outside world to inhibit the bistable when it wishes clocking to cease. The output waveform of the clock is sufficiently clean to use as a triggering device. However, as the Low voltage level may vary about -1.55 volts by (approximately) ±.05 volts, one may wish to put the waveform through a MC356 in Schmitt trigger configuration to clean it up further. This was not found necessary in the tape system. The tape clock was set for 20 microseconds, and used a capacitance of 2000 picofarads. The output polarities are as shown when the clock is inhibited.

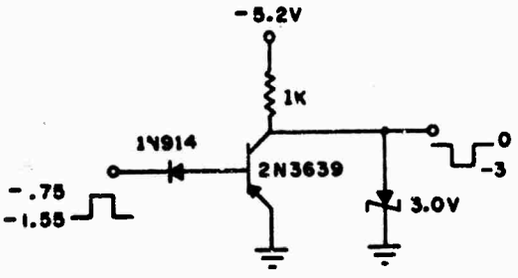


D. MECL to DEC Buffers.

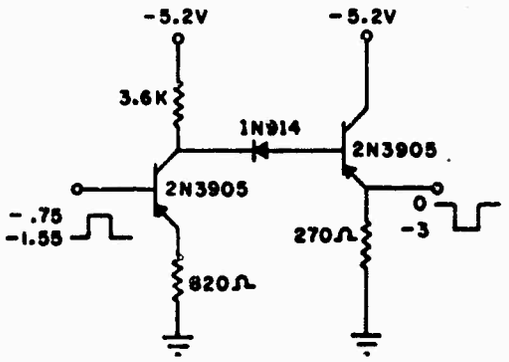
The one transistor buffer is essentially a "quick and dirty" means of converting a .8 volt swing to a 3 volt swing, using a 3 volt zener diode to tie down the output. Its chief disadvantages are the expense and variability of the zener diode, and the slow fall time (about 1 microsecond) due to emitter capacitance. Its advantage is that it requires only four elements, and thus can be mounted quickly and in high density on an ADC (Applied Development Corporation) circuit board (14 per board). The one transistor buffer can only be used in cases where the rise and fall times are not important.

The two transistor buffer is a better-designed device, also built by Thomas Chaney. It is a six element device which amplifies and inverts the MECL signal. The output will be at ground, or at -3 ±.1 volts, with a ten nanosecond rise and fall time. The disadvantage of this circuit lies in the increased number of elements and the resulting decrease in the number of buffers per board (10). As a result, this circuit was only used as the output to the Reader Writer card where the transition times are most critical.

One Transistor Buffer:



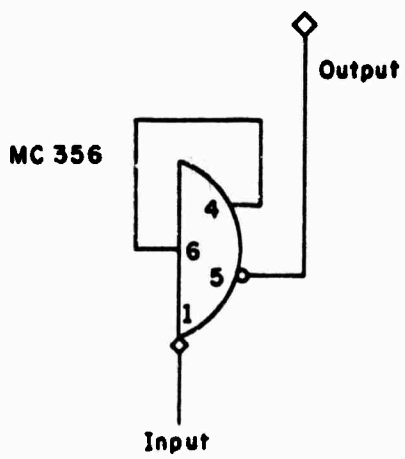
Two Transistor Buffer:



E. Schmitt Trigger.

This device was used as a DEC to MECL buffer interface. It has the advantages of high noise rejection, and standardization of both input signals and transition speeds. This device was created out of an MC-356, three input, dual-sexed output gate. The Schmitt trigger configuration is possible due to the existence of the bias input. This essentially makes a differential amplifier out of the 356. Thus, by feeding the signal in the bias driver input of the gate, and then comparing the amplified inverted output (feeding pin 4 into one of the standard gate inputs) to the initial signal, the amplifier will yield the differentially compared result at the output (either High or Low).

Because of the effect of emitter resistance, there is a .2 volt offset about the -1.15 volt mean. Hence, the device does not switch from Low to High until the input signal rises to nearly -.95 volts; and, likewise, it does not switch from High to Low until the signal drops to nearly -1.35 volts. Thus, as far as noise immunity is concerned, this is a desirable feature indeed. In switching from DEC to MECL levels (DEC level is 0 v. to -3.0 volts), one can tolerate over 2.5 volts peak to peak noise. Although this device has not been tested for maximum positive and negative voltage swings before breakdown, the trigger has been found to operate satisfactorily with an eight volt peak to peak swing around a -1.15 volt point.



APPENDIX IV: MAGNETIC TAPE PROGRAM (MTP)

```
      @200
#1A   STA  1          Save instruction and blk. #.
      0
      CLR           Turn off Linc bit.
      RTA           Place Quarter # in bits 0-2.
      SCR  3
      ADA  1        Form "ADD" instruction with address of correct ini-
      1C              tial index # for the indicated Quarter #.
      STC  1B
      ADD  2        Save Index Register "2".
      STC  7A+1
      ADD  1        Save Index Register "1".
      STC  7B
      ADD  0        Save Exit.
      STC  7C
      ADD  1A+1     Place instruction and blk. # in Acc.
      COM  ┐
      ROR  6  │     This routine forms an "LDA i1" instruction if we are
      BCL  1  │     doing a WRT instruction, or an "STA i1" if we are
      7737    }     doing a Read instruction, and stores proper in-
      BSE  1  │     struction in routine.
      LDA  i1 │
      STC  1F ┘
#1H   CLR           Turn off Linc bit.
      ADD  1A+1     Place instruction and blk. # in Acc.
      OPR  3        Execute instruction.
      SET  1        Load IR-1 with proper Initial Quarter Address.
#1B   0
      SET  i2       Store length of block in IR-2.
      -400
#1E   SXL  i4
      JMP  1F-1     Exit if TPRDY.
      SXL  i2
      JMP  7D       Exit if Fail.
      SXL  i6
      JMP  1G+2     Exit if Done.
      JMP  1E       Recycle until one of 3 options is satisfied.
      OPR  i4       Read into A register (if RTP instruction).
#1F   LDA  i1       Execute transfer instruction.
      OPR  i5       Give OK, wait for TPRDY.
      XSK  i2       Index block length register.
      JMP  1E       Transfer next word.
#1G   SXL  i6       Test for done level.
      JMP  7E       Done!  Go test for possible failure.
      SXL  i2       Test for failure.
      JMP  7D       Failed!  Go test for WRC or RDC instruction.
      JMP  1G       Cycle until one of two options is satisfied.
```

```
#7D CLR
    COM        Turn on Linc bit (Fail Indicator).
    SCR i1
    LDA        Place instruction and blk. # in Acc.
    1A+1
    ROL 2      Test for RDC or WRC instruction.
    APO i
    JMP 1B-1   If RDC or WRC instruction, return and try again.
#7A LDA i
    0          Restore Index register 2.
    STC 2
    LDA i
#7B 0          Restore Index register 1.
    STC 1
#7C 0          Return to main routine.
#7E SXL i2     Test for failure.
    JMP 7D     Failed! Jump to Fail indicator routine.
    JMP 7A     O.K., Jump to Exit!
#1C 1777       Table of initial settings:  Quarter "0"
    0377                                   Quarter "1"
    0777                                   Quarter "2"
    1377                                   Quarter "3"
    3777                                   Quarter "4"
    2377                                   Quarter "5"
    2777                                   Quarter "6"
    3377                                   Quarter "7"
```

APPENDIX V: MODERN MODULE MARK ROUTINE (MODMRK)

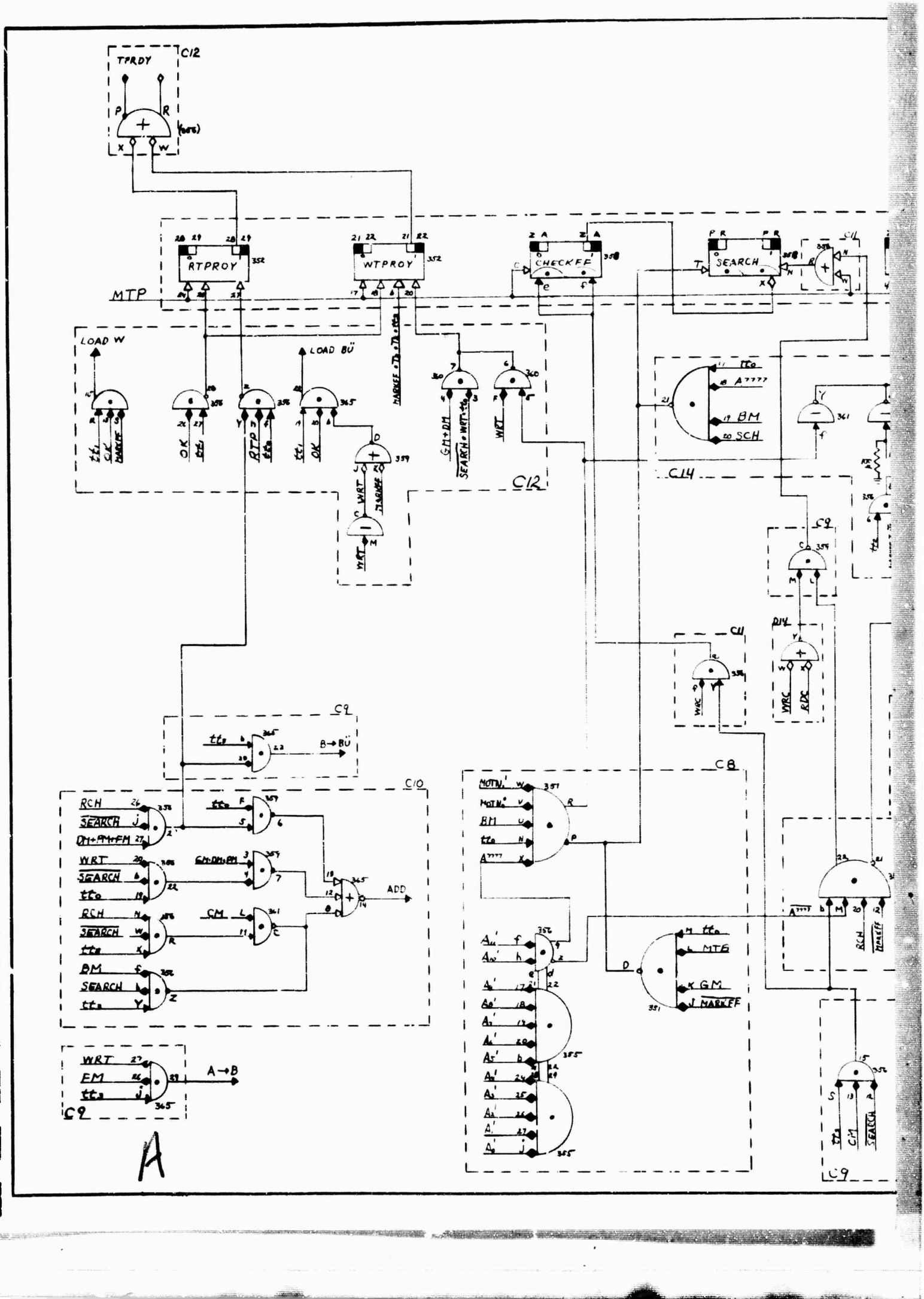
```
          @20
          MTB i          Rewind Program tape on unit "0".
          0
          JMP 6A         Display Comment "A".
          @40
          SET i3         Set I.R. 3 to # blks. to be written.
          -1023
          SET i4         Forward blk. #. Begin with -10 (complemented) for
          10               directional purposes.
          SET i6         Set counter for initial IBZ length.
          -10
          LDA i          Place Mark Code plus unit "1" selection in Acc.
          6002           Put unit "1" in window.
          ATR
          OPR 7          MARK!
          CLR
          COM            Place IM Mark in window (bits 0-3).
          ATR
#1G       SET i12        IM counter.
          0
#1A       OPR i4         Wait for TPRDY (write IM).
          OPR i5         Wait for TPRDY and give OK level.
          XSK i12        Index.
          JMP 1A
          XSK i6         Are we through IBZ?
          JMP 1G         No, write some more!
#1D       LDA i
          16             Place BM in window.
          ATR
          OPR i4         Wait for TPRDY.
          LDA
          4              Place FWD blk. # in Acc. (write BM).
          OPR i5         Set OK and wait for TPRDY.
          LDA i
          11             Place GM in window.
          ATR
          OPR i4         Write GM.
          OPR i5
          LDA i
          1              Place DM in window.
          ATR
          SET i7
          0              Clear BWD blk. # register.
          OPR i4         Wait for TPRDY.
          LDA i
          7777           Place all "1"'s in alternate data words.
          OPR i5         Give OK, wait for TPRDY (write 1st DM).
          LDA
          4              Place FWD blk. # in Acc.
          BCL i          Mask off all but 1st 4 bits (0-3).
          7760
          ADA i          ADD Table address of Reverse complemented equiv-
          7Z               alent of the 4 bit #.
          STC *+5
          OPR i4         Wait for TPRDY.
          CLR            CLR alternate Data words (write 2nd DM).
```

```
        OPR i5          Give OK, wait for TFRDY.
        LDA
        0
        LAM             Place Reverse complemented equivalent # in
        7                 corresponding bits of BWD blk. #.
        OPR i4
        LDA i           Place all "1"'s in Acc. (write 3rd DM).
        7777
        OPR i5
        LDA             Place FWD blk. # in Acc.
        4
        BCL i           Mask off all but 2nd. 4 bits (4-7).
        7417
        ROR 4           Normalize right.
        ADA i           ADD address of Table of Reverse complemented
        7Z                equivalent #'s.
        STC p+5
        OPR i4          Write 4th data mark.
        CLR
        OPR i5
        LDA             Load Acc. with reverse complemented equivalent
        0                 of 2nd 4 bits.
        ROL 4           Normalize left.
        LAM
        7               Set 2nd. 4 bits of BWD #.
        OPR i4
        LDA i           Write 5th data mark.
        7777
        OPR i5
        LDA             Load Acc. with FWD Blk. #.
        4
        BCL i           Mask off all but last 4 bits (8-11).
        0377
        ROR 10          Normalize right.
        ADA i           ADD address of table of reverse complemented
        7Z                equivalent #'s.
        STC p+5
        OPR i4
        CLR             Write 6th data mark.
        OPR i5
        LDA             Load Acc. with reverse complemented equivalent
        0                 of last 4 bits.
        ROL 10          Normalize left.
        LAM             Set last 4 bits of reverse complemented FWD blk. #
        7                 (i.e., the BWD blk. #).
        OPR i4
        LDA i           Write 7th data mark.
        7777
        OPR i5
        SET i12         Write the remaining 367 DM's (there are 376 in all),
        -174              which means cycle through the following routine:
#1C     OPR i4          (367+1)/2=174 times (octal).
        CLR             CLR alternate data words (even words).
        OPR i5
        XSK i12         Are we through?
        JMP p+2         No!  Write another odd word.
```

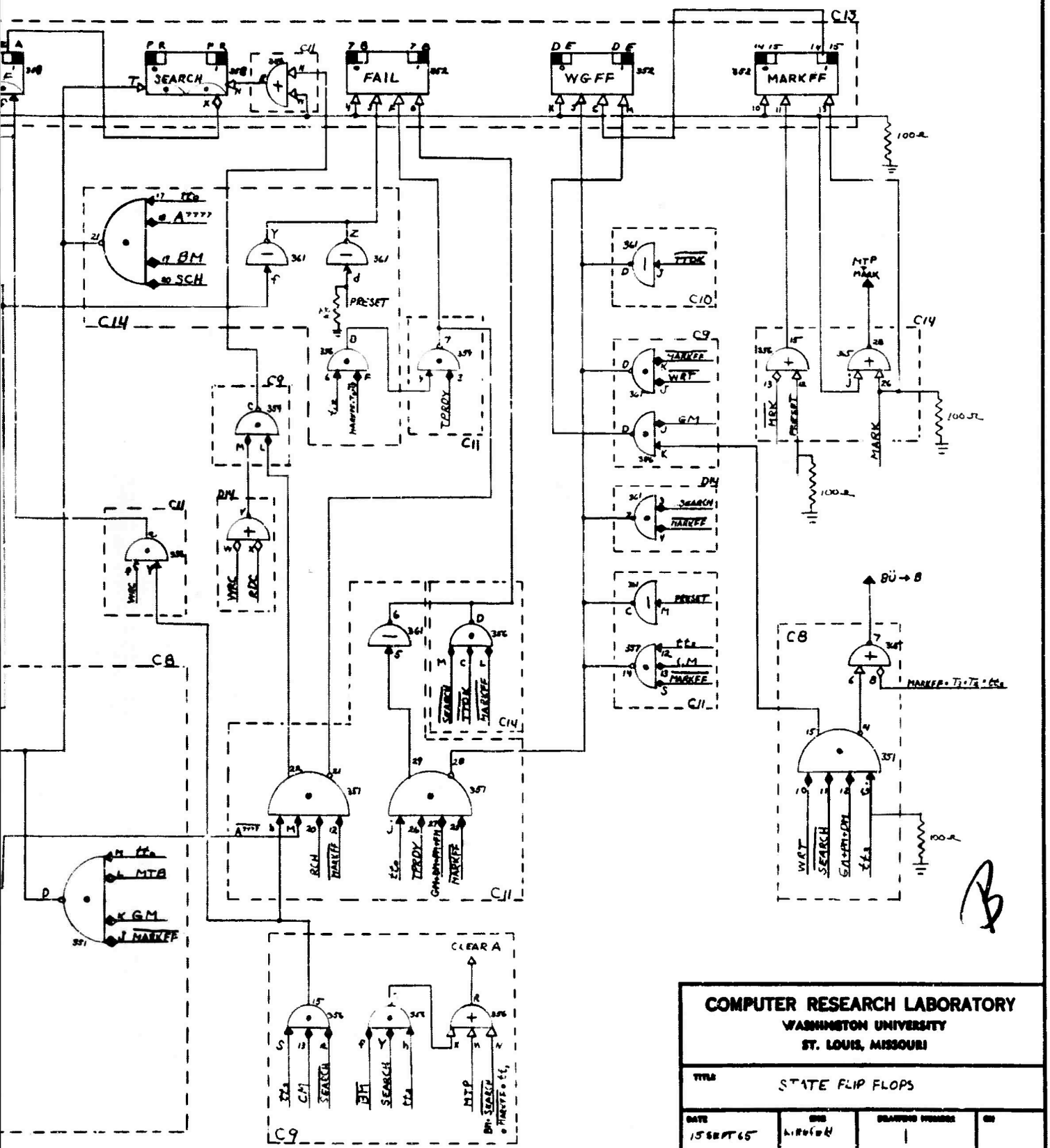
```
        JMP  P+6        Through!  Continue...
        OPR  i4
        LDA  i          Write an odd data word.
        7777
        OPR  i5
        JMP  1C
        LDA  i
        13              Place Prefinal mark in window (PM).
        ATR
        OPR  i4
        LDA  i          Write Prefinal mark and odd data word.
        7777
        OPR  i5
        LDA  i
        5               Place Final Mark in window (FM).
        ATR
        OPR  i4
        CLR             Write even FM.
        OPR  i5
        CLR             Place CM in window.
        ATR
        OPR  i4
        LDA  i          Write the correct check-sum complement (i.e.,
        0177              two's complement of 200(8) complemented).
        OPR  i5
        OPR  i4
        OPR  i5         Write 2nd CM.
        OPR  i4
        OPR  i5         Write 3rd CM.
        LDA  i
        7               Place BM in window.
        ATR
        OPR  i4
        LDA
        7               Place BWD blk. # in Acc.
        OPR  i5         Write BM.
        CLR
        COM             Place IM in window.
        ATR
        OPR  i4         Write 1st IM.
        OPR  i5
        LDA
        4
        ADA  i          Index FWD blk. #.
        -1
        STC  4
        OPR  i4         Write 2nd IM.
        OPR  i5
        SET  i12
        -3
        OPR  i4         Write remaining IBZ.
        OPR  i5
        XSK  i12
        JMP  P-3
        XSK  13         Are we through?
        JMP  1D         No, write another block.
```

```
        SET i12         Through, write a long IBZ at end of tape.
        0
#1E     OPR i4
        OPR i5
        XSK i12
        JMP 1E
        LDA i           Through marking, reverse motion for recheck:
        3               Place unit option and 1-bit option in window.
        ATR
        SET i1          Set counter for entire tape length.
        -1000
        SET i2          Clear blk. # register.
        0
#7A     LDA i           Place check instruction in Acc.
        3000
        ADD 2           Add block #.
        OPR 3           MTPP!  Leave marking mode, go into MTP mode.
        OPR i6          Wait for DONE level.
        SXL i2          Skip it, no fail level.
        JMP 7B          Failed, display comment.
        XSK i2          Index blk. #.
        XSK i1          Are we done?
        JMP 7A          No, check another block.
        LDA i           DONE, Place MTB "0" in Acc.
        7000
        OPR 3           Rewind tape.
        JMP 6B          Display comment "B", "GOOD TAPE".
#7B     LDA i           FAILED, Place MTB "0" in Acc.
        7000
        OPR 3           Rewind tape.
        JMP 6C          Display comment "C", "BAD TAPE, TRY AGAIN".
#7Z     0017            REVERSE COMPLEMENTED EQUIVALENT TABLE:
        0007
        0013
        0003
        0015
        0005
        0011
        0001
        0016
        0006
        0012
        0002
        0014
        0004
        0010
        0000
#6A     JMP 1000        Display comment "A".
        1414
        1414
        1414
        4042
        2740
        4536
        1412
        1224
```

```
      3737
      4252
      1447
      2443
      3014
      4742
      1445
      3052
      3441
      2712
      4337
      2426
      3014
      2537
      2441
      3614
      4724
      4330
      1442
      4112
      5041
      3447
      1401
      1230
      4132
      2432
      3014
      2537
      3441
      2630
      1246
      4724
      4547
      1404
      0077
      JMP 6A
#6B   JMP 1000      Display Comment "B".
      3242
      4227
      1447
      2443
      3077
      JMP 6B
#6C   JMP 1000      Display Comment "C".
      2524
      2714
      4724
      4330
      6014
      4745
      5414
      2432
      2434
      4177
      JMP 6C
```



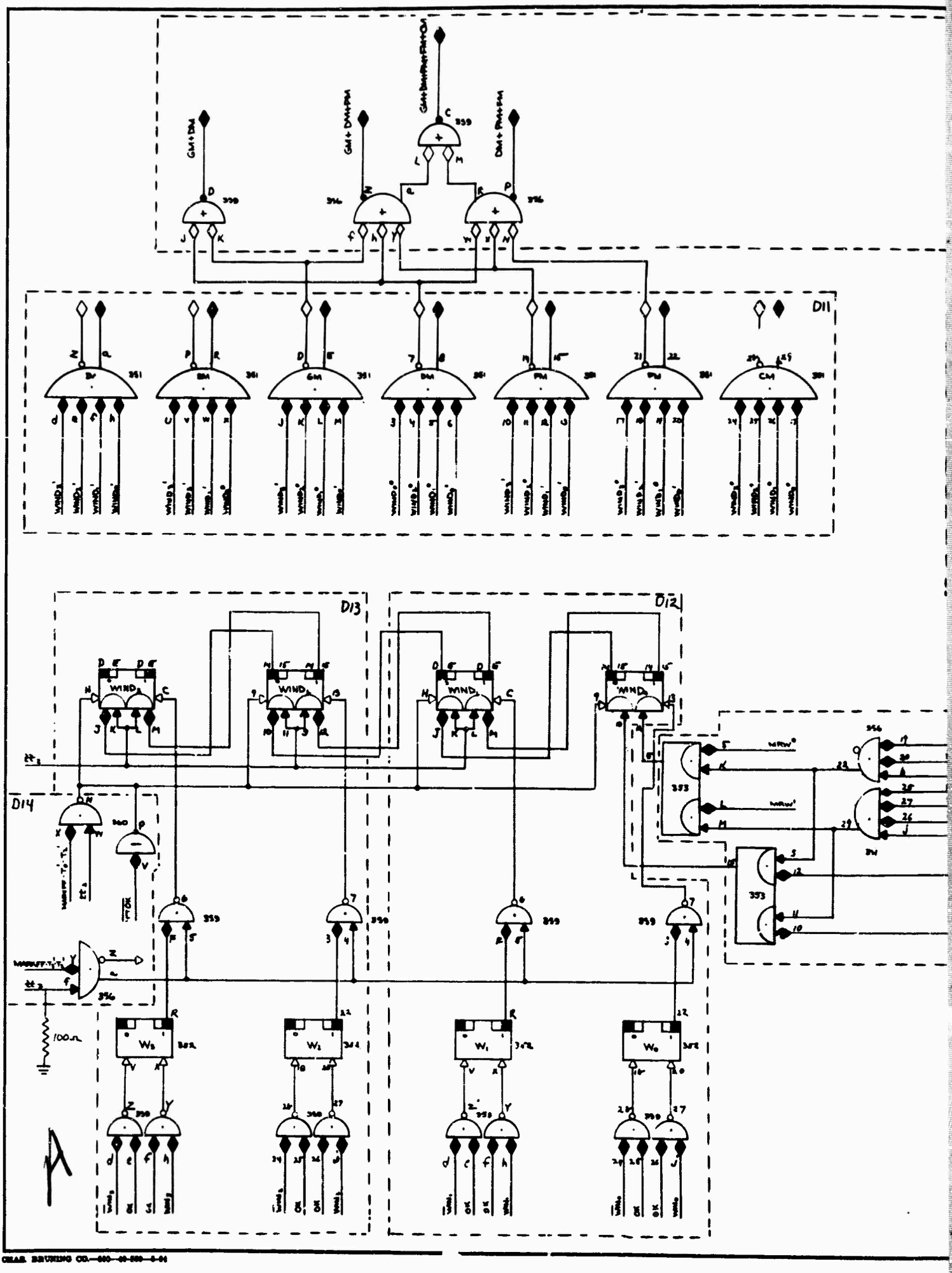
TPRDY
C12
RTPROY
WTPROY
CHECKFF
SEARCH
MTP
LOAD W
LOAD BÜ
OK
RTP
WRT
GM+DM
SEARCH
C12
C14
BM
SCH
C9
C11
WRC
RDC
B→BÜ
C10
RCH
SEARCH
DM+PM+FM
WRT
GM+DM+PM
CM
ADD
BM
C8
MTB
GM
MARKFF
FM
A→B
A

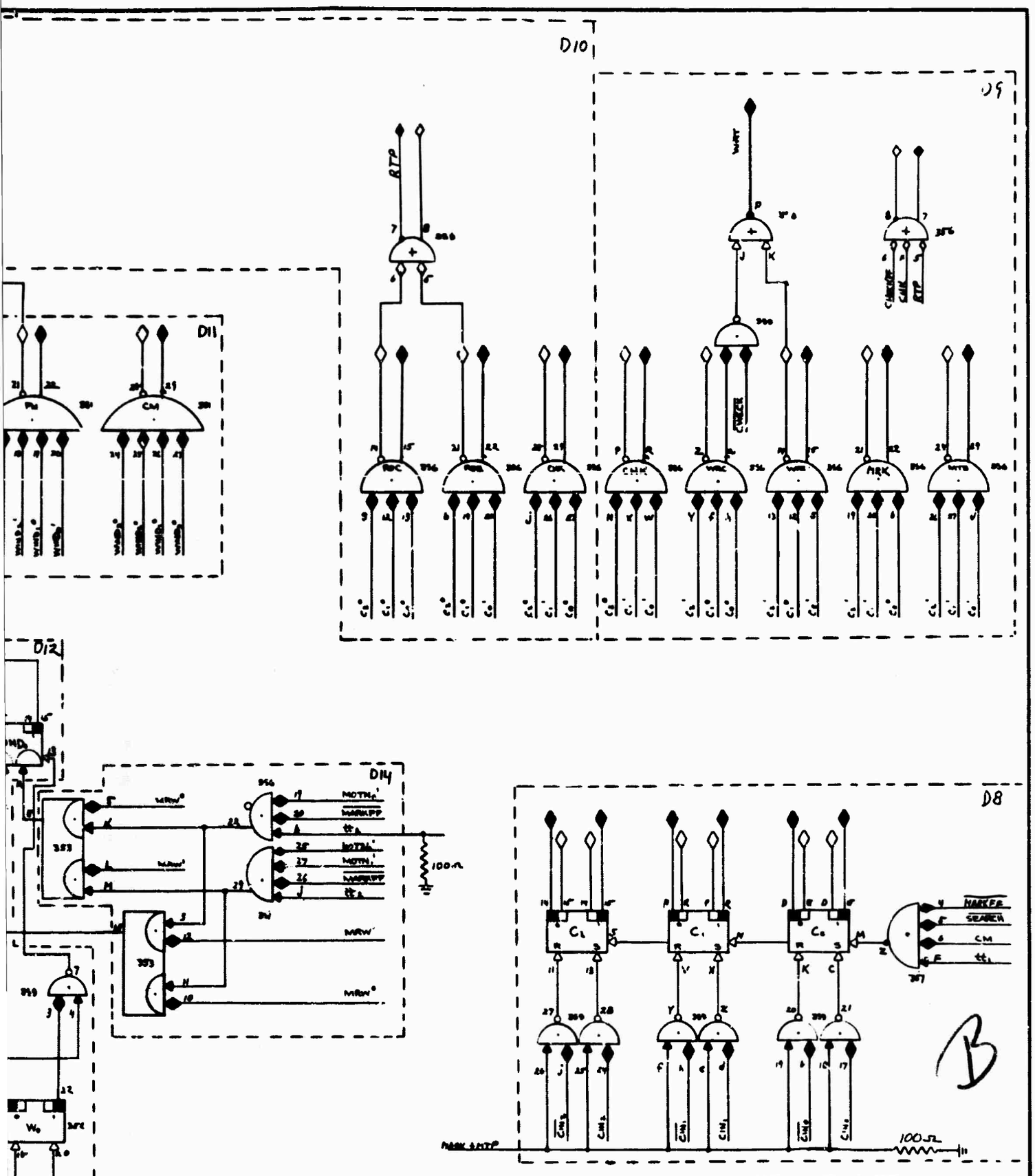


COMPUTER RESEARCH LABORATORY
WASHINGTON UNIVERSITY
ST. LOUIS, MISSOURI

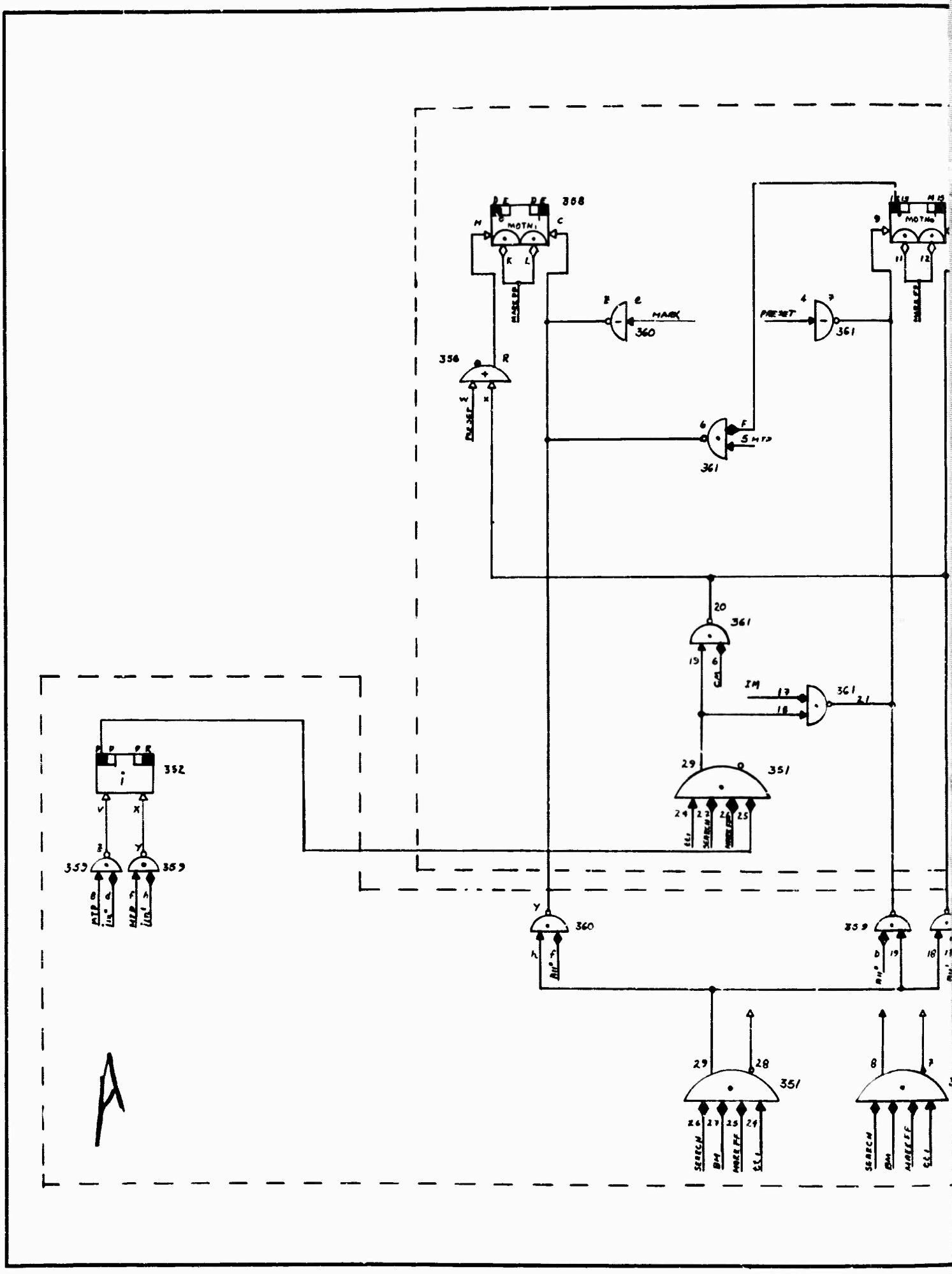
TITLE: STATE FLIP FLOPS

DATE	DWG	DRAWING NUMBER	REV
15 SEPT 65		1	

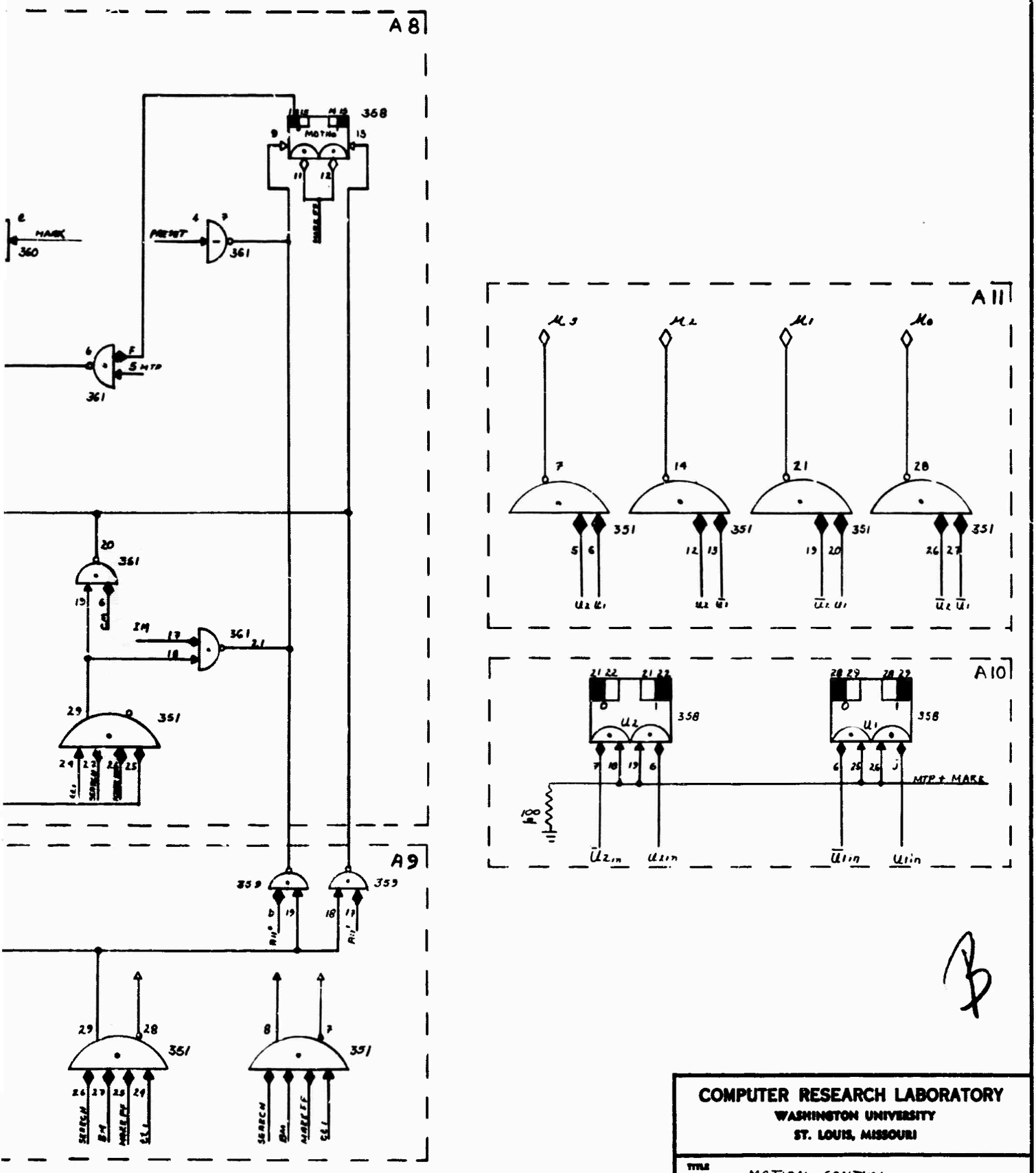

D11
D12
D13
D14
100Ω



BIOMEDICAL COMPUTER LABORATORY	
W. LITTLEFIELD	DRAWN BY HGL
	APPROVED BY
TITLE: DECODERS	
DATE: 15 SEP65	DRAWING NUMBER: 2

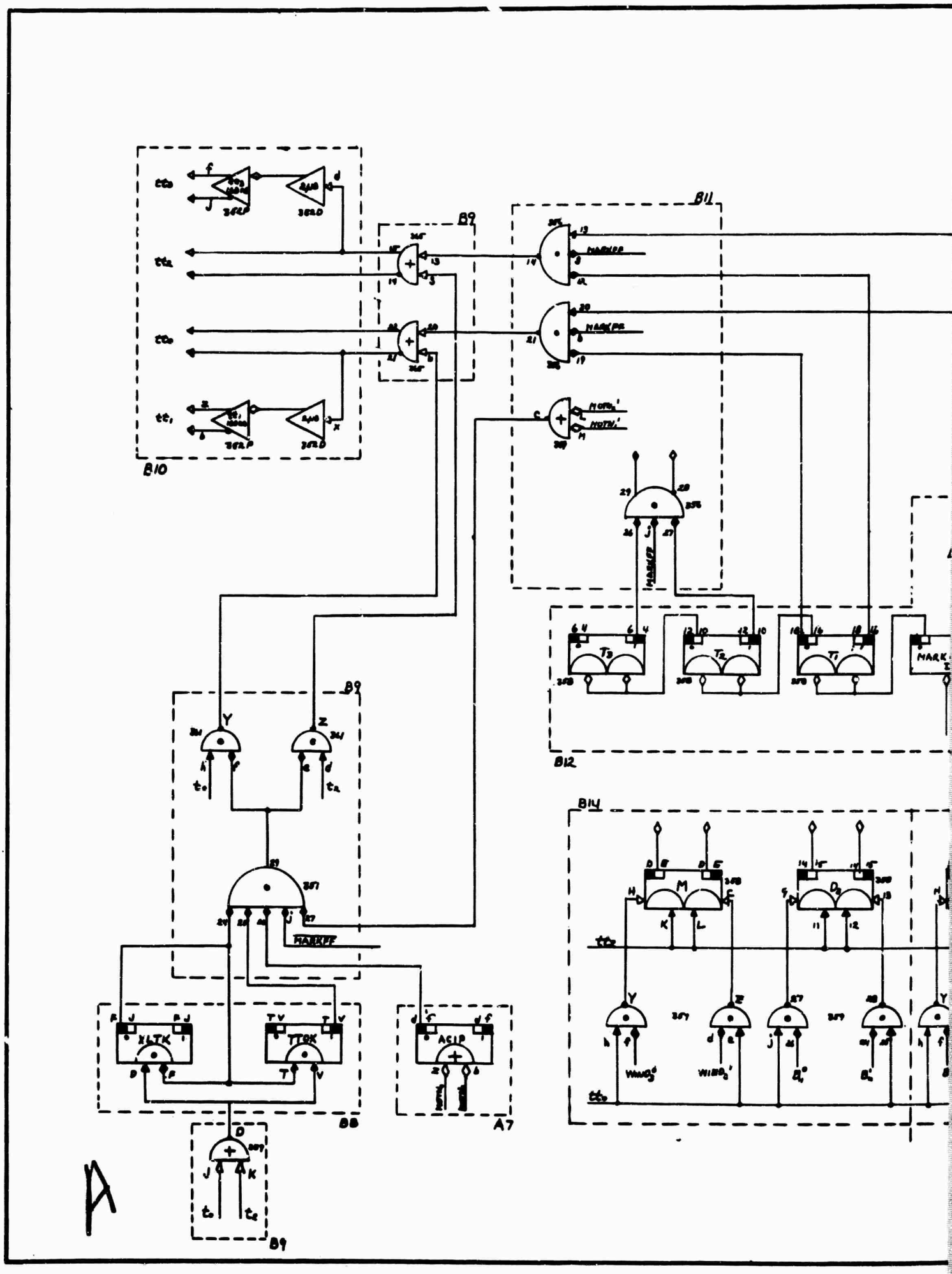


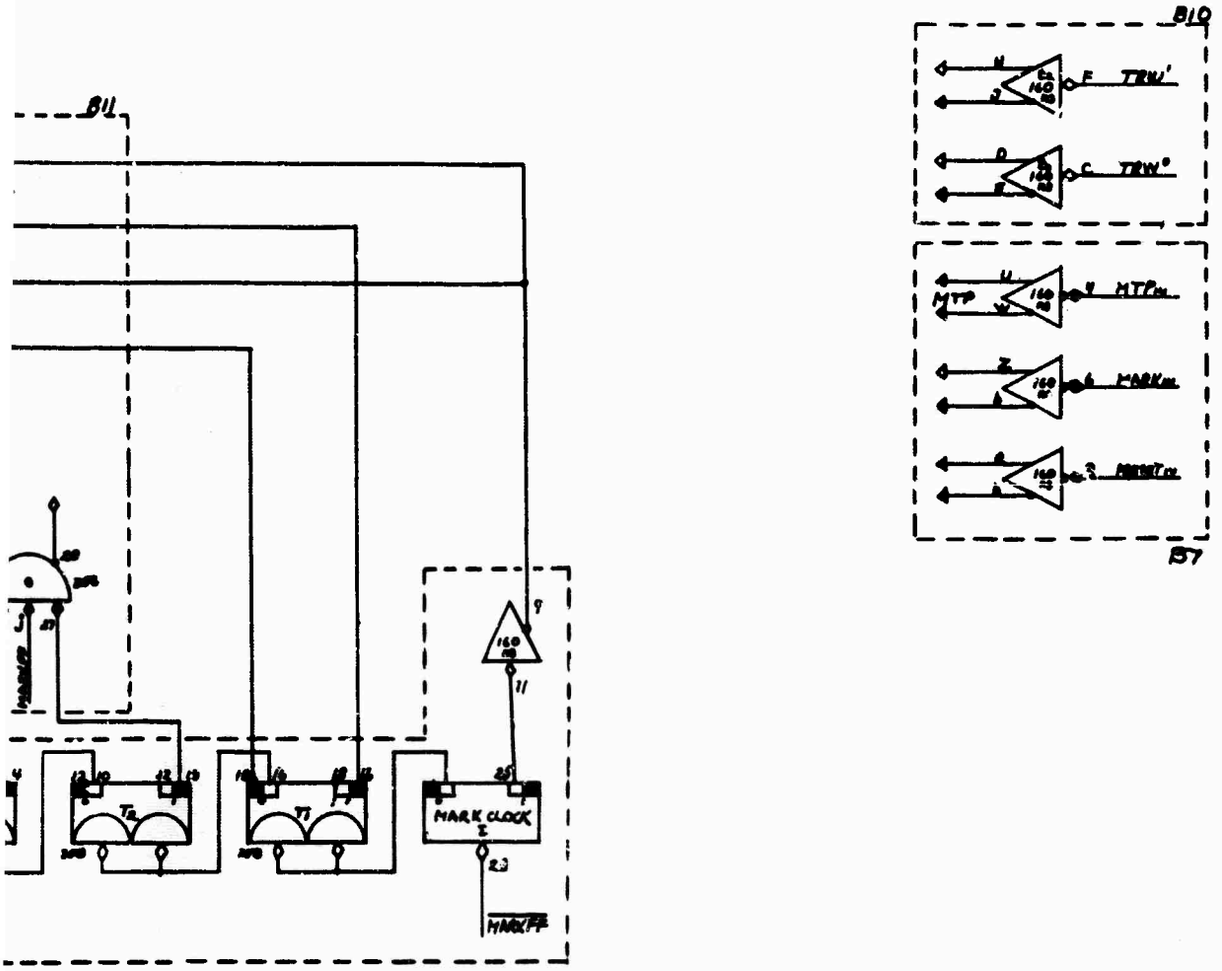
358
MOTN1
MOTN0
MARK
360
PRESET
361
356
5 MTD
361
20
361
IM
361
29
351
352
359
359
360
359
29
28
351
8
7
SEARCH
BM
MARK FF
CC1
A

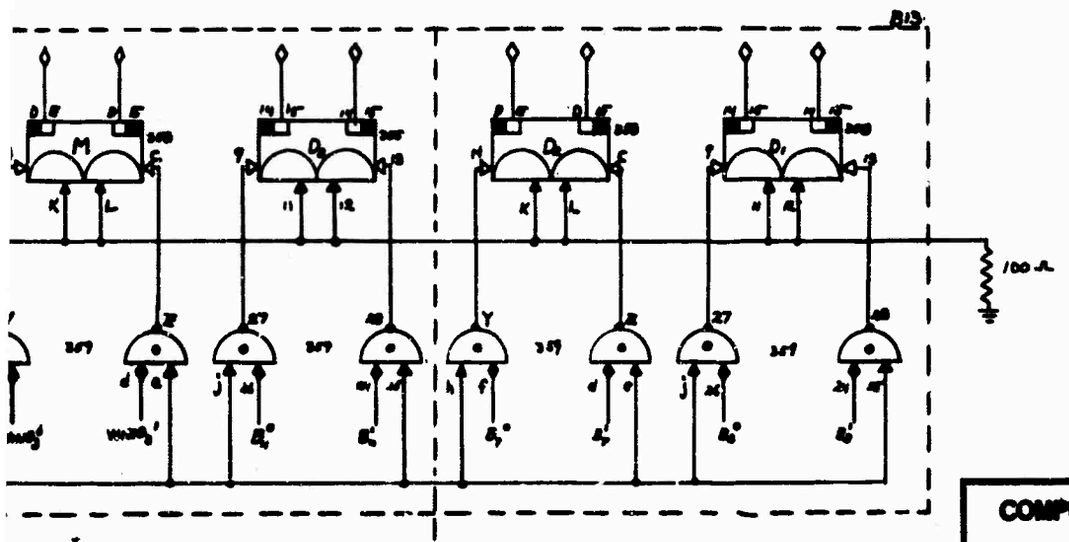


B

COMPUTER RESEARCH LABORATORY WASHINGTON UNIVERSITY ST. LOUIS, MISSOURI			
TITLE: MOTION CONTROL			
DATE: 15 SEPT 1965	DRG: LITTLEFIELD	DRAWING NUMBER: 3	REV

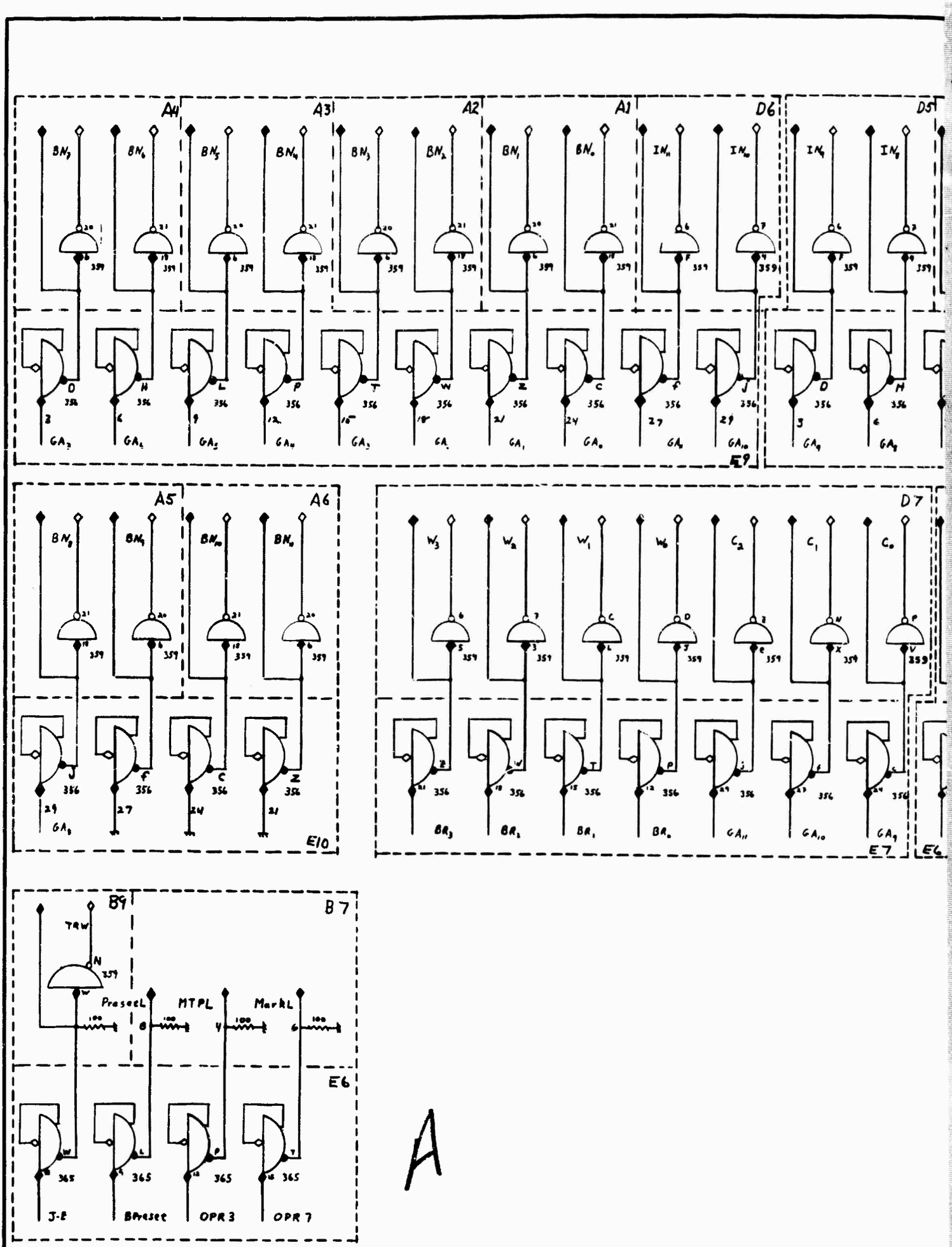

B10
B9
B11
B12
B14
B8
A7
B9
MARKFF
XLTK
TTOK
ACIP
A



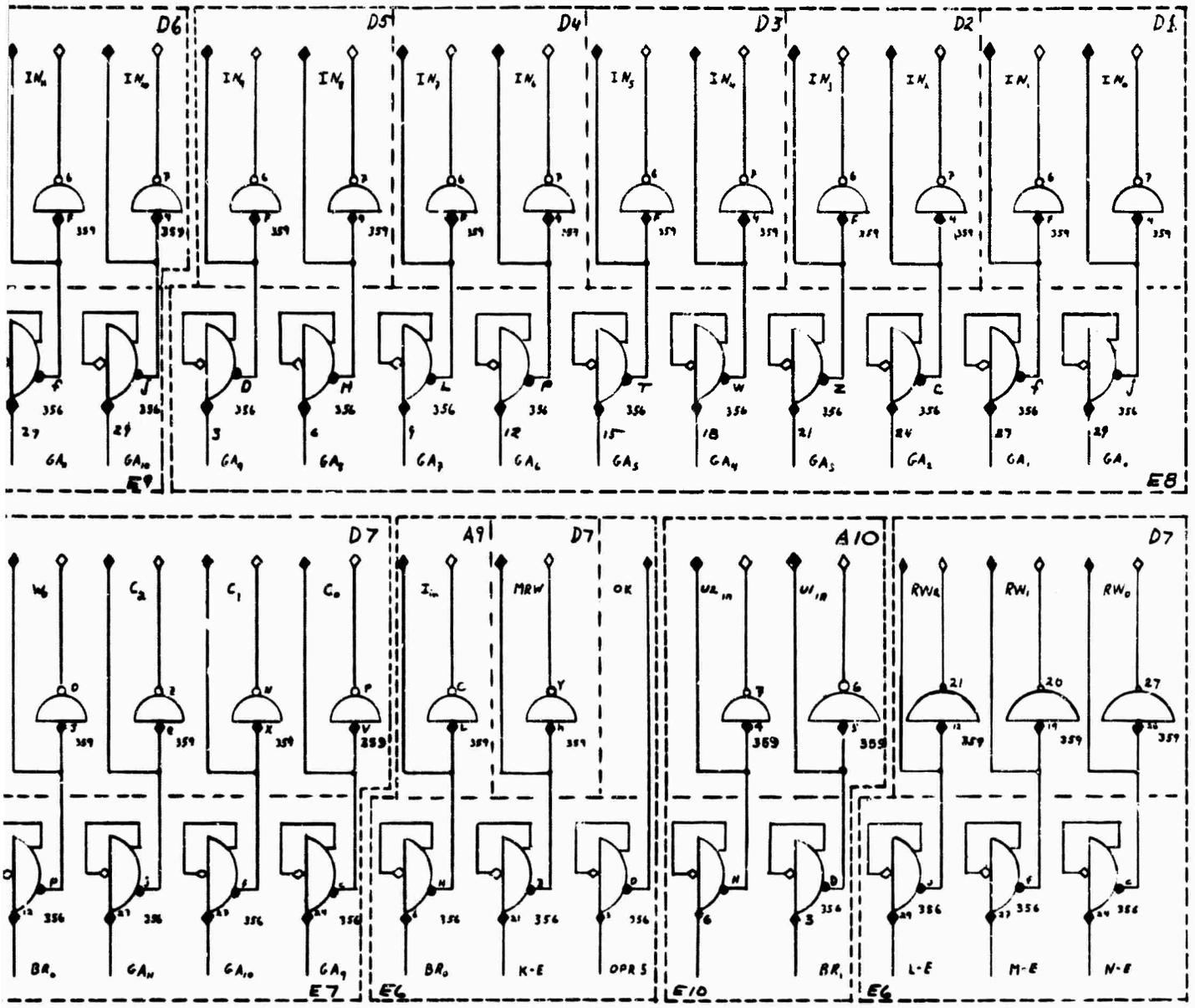


B

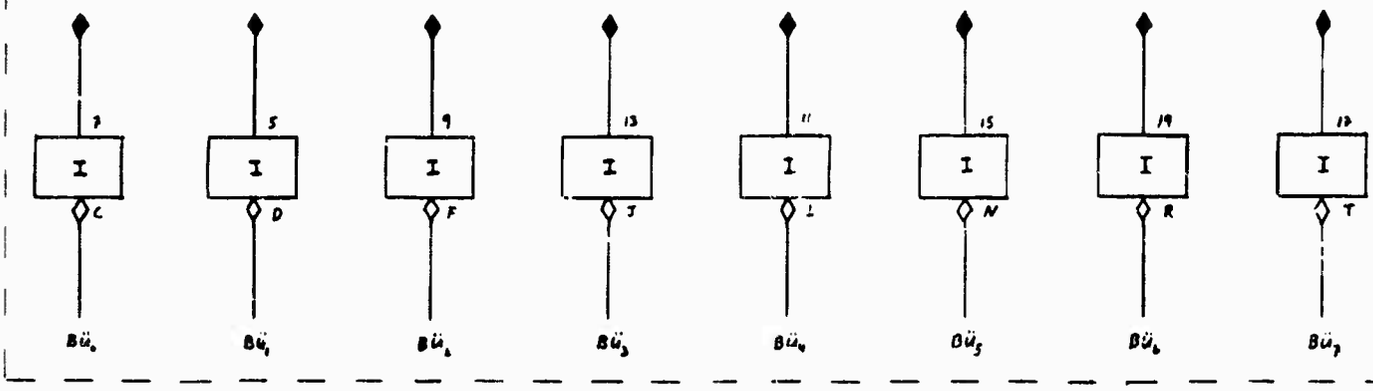
COMPUTER RESEARCH LABORATORY WASHINGTON UNIVERSITY ST. LOUIS, MISSOURI			
TITLE: MARK, TIMING & DELAYS			
DATE: 19 SEPT 1965	DRN: LITTLEFIELD	DRAWING NUMBER: 4	CH

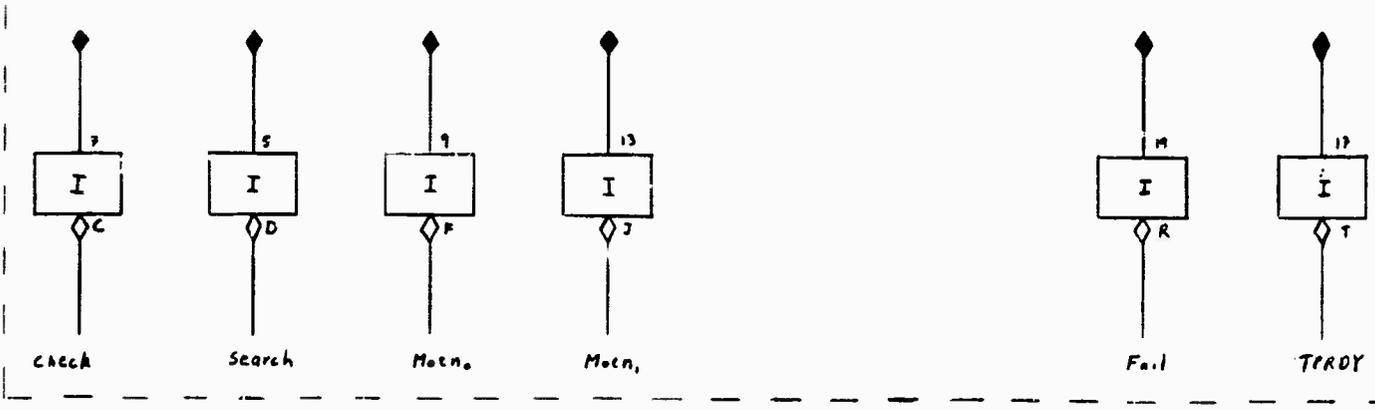

A4
A3
A2
A1
D6
D5
A5
A6
D7
B9
B7
E9
E10
E7
E6
TRW
PresetL
MTPL
MarkL
J-E
BPreset
OPR 3
OPR 7

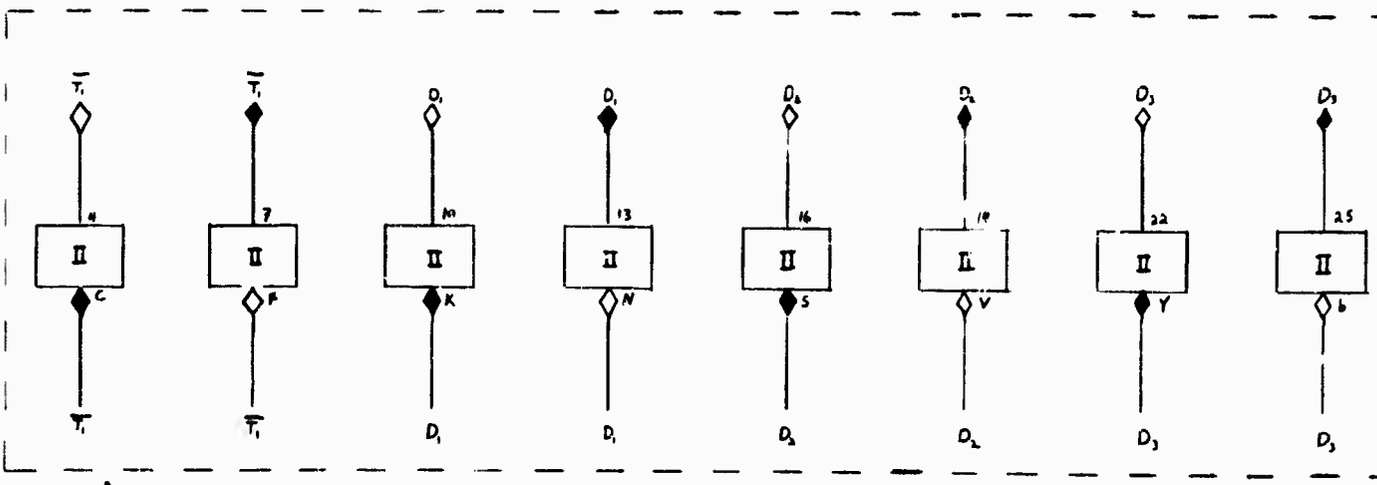
A



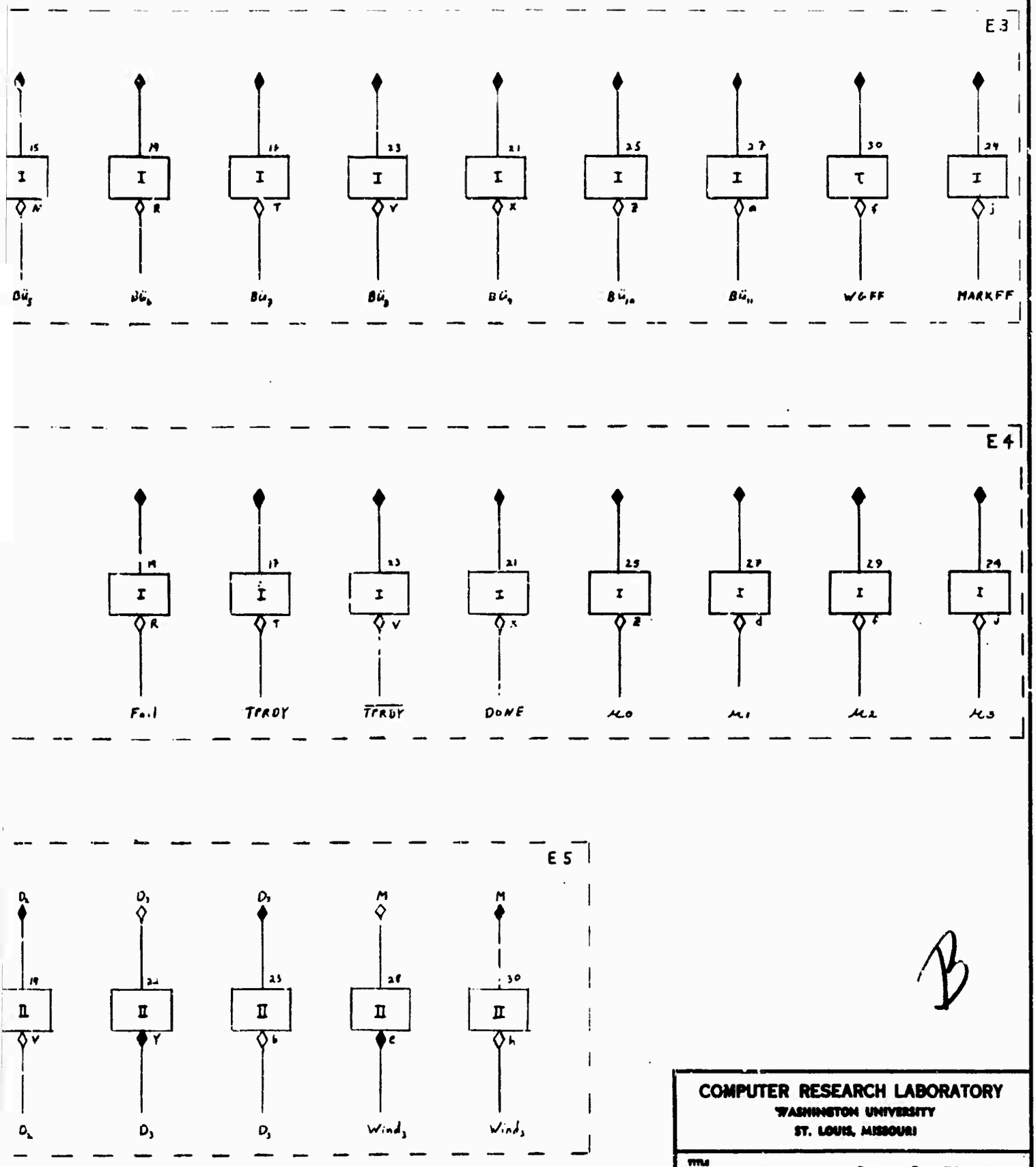
COMPUTER RESEARCH LABORATORY WASHINGTON UNIVERSITY ST. LOUIS, MISSOURI			
TITLE: DEC to MECL Buffers			
DATE: 9/9/66	DRN: WML	DRAWING NUMBER: 5	REV:


3
I
C
BÜ0
5
I
D
BÜ1
9
I
F
BÜ2
13
I
J
BÜ3
11
I
L
BÜ4
15
I
N
BÜ5
19
I
R
BÜ6
17
I
T
BÜ7


3
I
C
check
5
I
D
Search
9
I
F
Mem0
13
I
J
Mem1
M
I
R
Fail
17
I
T
TRDY


T1
4
II
C
T1
T1
7
II
F
T1
D1
10
II
K
D1
D1
13
II
N
D1
D2
16
II
S
D2
D2
19
II
V
D2
D3
22
II
Y
D3
D3
25
II
b
D3

A

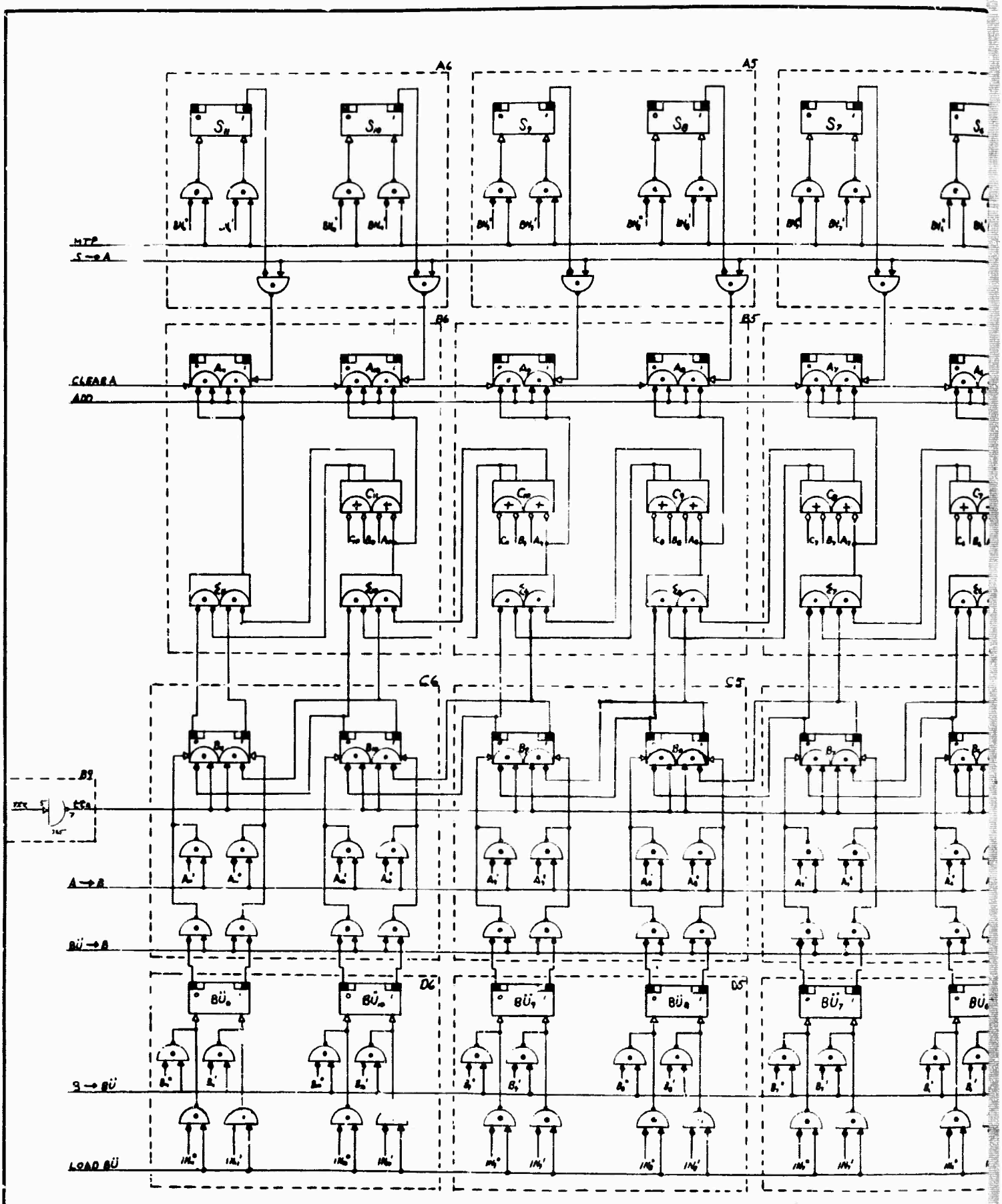


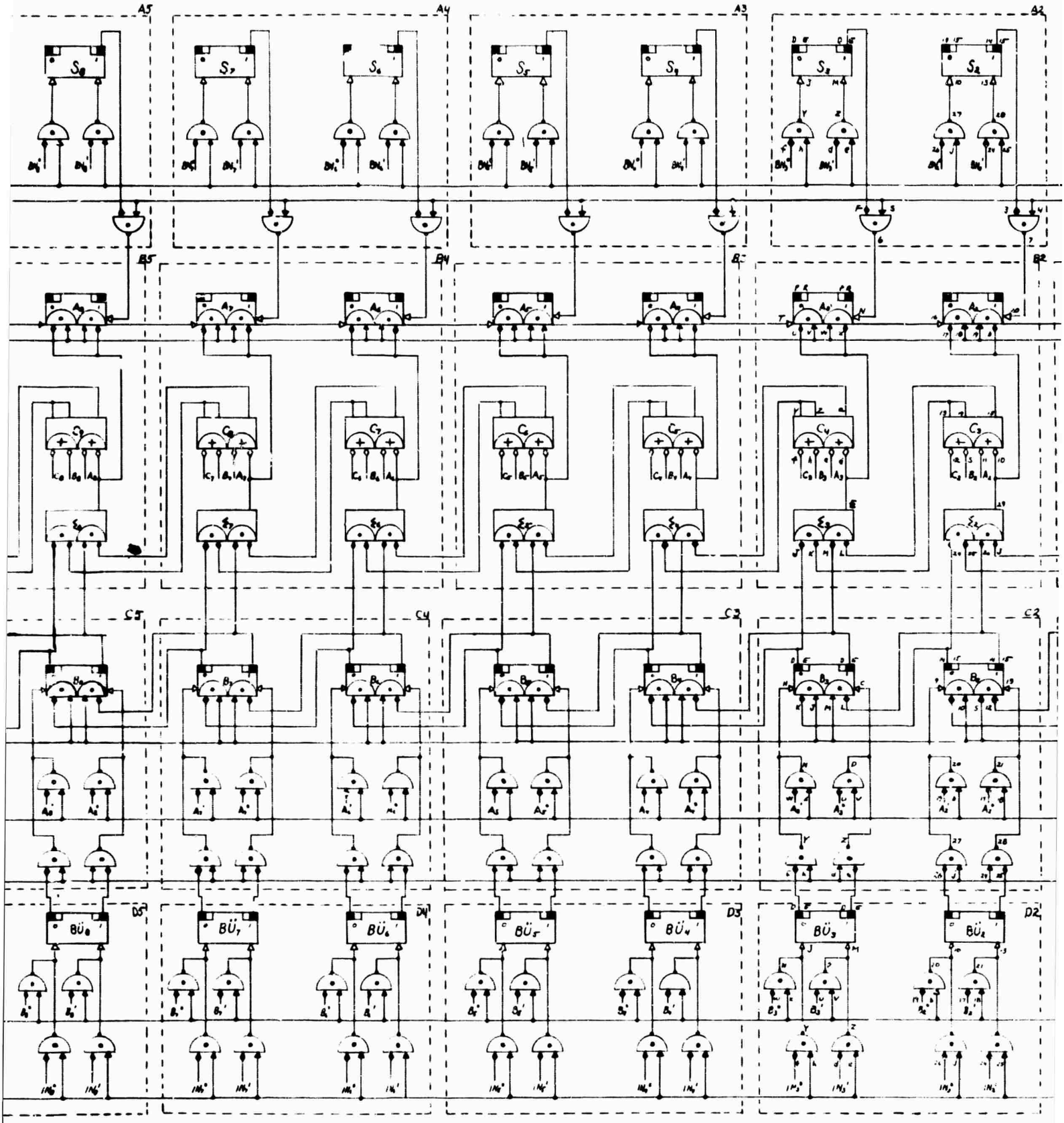
B

COMPUTER RESEARCH LABORATORY
WASHINGTON UNIVERSITY
ST. LOUIS, MISSOURI

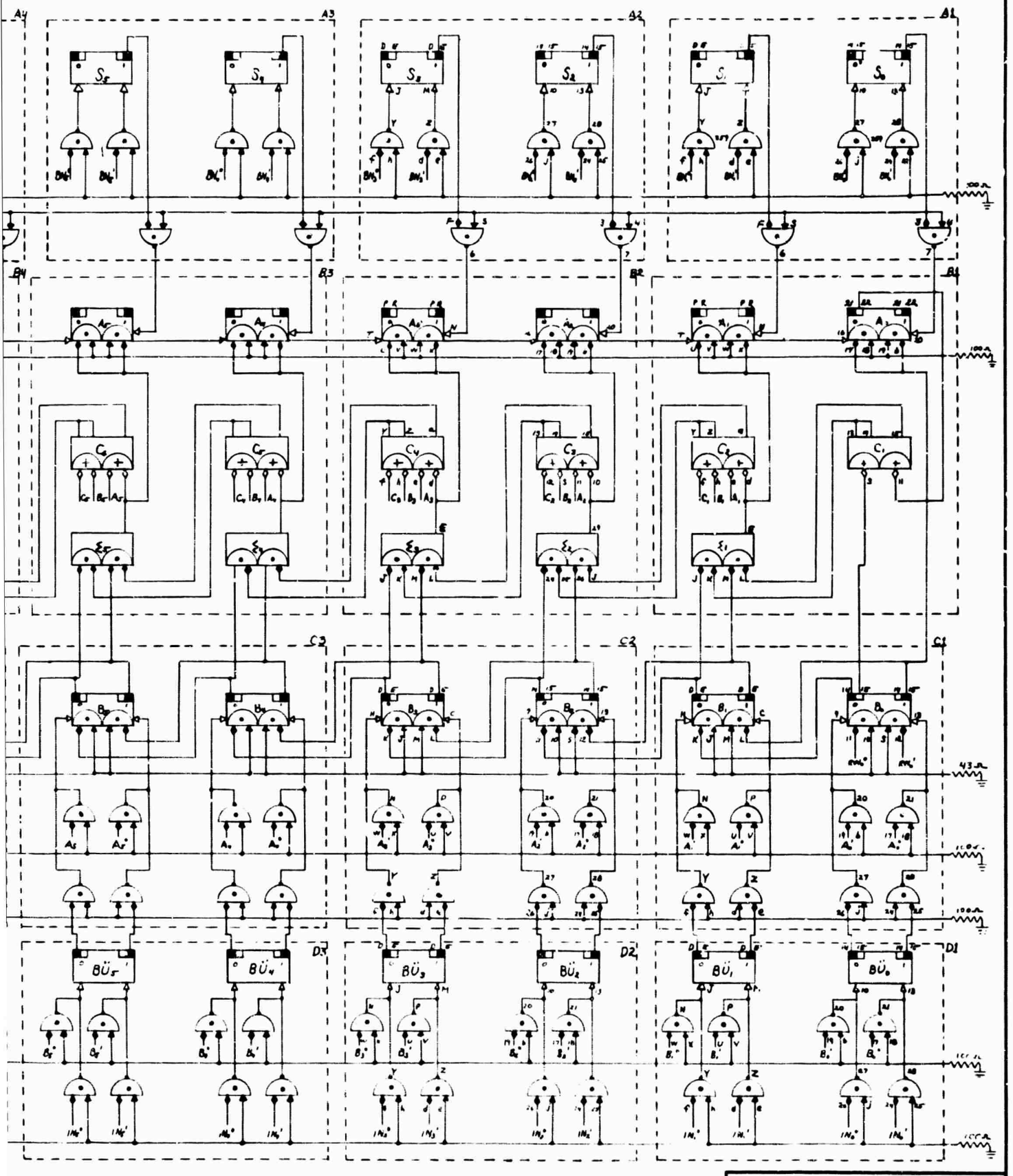
TITLE: MECL to DEC BUFFERS

DATE	DWG	DRAWING NUMBER	CH
9/12/66	[illegible]	6	


A6
A5
B6
B5
C6
C5
D6
D5
B8
HTP
S → A
CLEAR A
ADD
A → B
BÜ → B
9 → BÜ
LOAD BÜ



B



COMPUTER RESEARCH LABORATORY
WASHINGTON UNIVERSITY
ST. LOUIS, MISSOURI

TITLE: DATA REGISTER

DATE | DRN LITTLEFIELD | DRAWING NUMBER | SH

BIBLIOGRAPHY

Clark, Wesley and Molnar, Charles. "The LINC", Proceedings, New York Academy of Sciences Conference on Computers in Medicine and Biology, (May 27-29, 1963).

Clark, Wesley A.; Stucki, Misheli J., and Ornstein, Severo M. "A Macromodular Approach to Computer Design", St. Louis: Washington University, 1966.

Ornstein, Severo and Molnar, Charles. "LINC Theory of Operations II", (St. Louis: Washington University, 1966), XIV.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Computer Systems Laboratory,
Washington University
724 South Euclid, St. Louis, Missouri 63110

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
The Design of a Tape Macromodule

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Interim-Technical

5. AUTHOR(S) *(First name, middle initial, last name)*
Warren M. Littlefield

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 15, 1967	78	3

8a. CONTRACT OR GRANT NO.
(1) DOD (ARPA) Contract No. SD-302
(2) NIH (DRFR) Grant No. FR-00218
b. PROJECT NO.
ARPA Project Code No. 5880
c. Order No. 655
d.

9a. ORIGINATOR'S REPORT NUMBER(S)
Technical Report No. 3

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
ARPA-Information Processing Techniques,
Washington, D.C.,
NIH Div. of Research Facilities

13. ABSTRACT

This report deals with the design and function of a Magnetic Tape System Module. The prototype was built out of MECL integrated logic and used in the pulse manner, and handles all those functions peculiar to the basic LINC tape transport. The system was interfaced and debugged on a LINC computer which was programmed to behave as a macromodular interface. In order to achieve a complete understanding of the material contained in this report, it is necessary that one be thoroughly familiar with the general theory of tape operation as contained in the first six chapters of the "LINC Theory of Operations II, Volume 14."

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Tape Macromodule						
Macromodule						
Marking Code						
Data Channel Structure						
Search Phase						
State Flipflops						
Read-Check Phase						
Write Phase						
Data Registers						
Timing Logic						
Testing Procedure						